

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

**THE DESIGN AND IMPLEMENTATION
OF AN AUTOMATED INTELLIGENCE
COLLECTION MANAGEMENT TOOL**

by

Karen Denyse Hensley Tomlin

September 1995

Thesis Advisor:

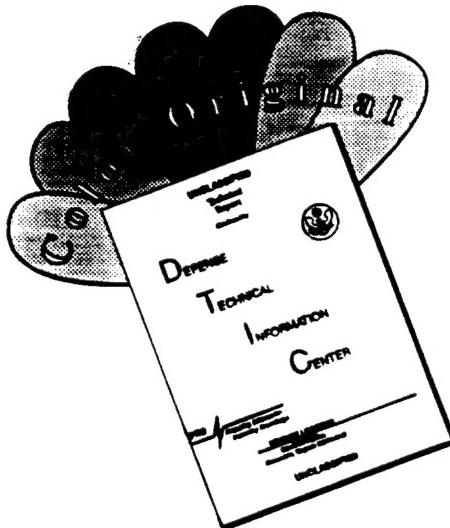
David R. Pratt

Approved for public release; distribution is unlimited.

DNC QUALITY INSPECTED

19960208 104

DISCLAIMER NOTICE



**THIS DOCUMENT IS BEST
QUALITY AVAILABLE. THE COPY
FURNISHED TO DTIC CONTAINED
A SIGNIFICANT NUMBER OF
COLOR PAGES WHICH DO NOT
REPRODUCE LEGIBLY ON BLACK
AND WHITE MICROFICHE.**

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time reviewing instructions, searching existing data sources gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)			2. REPORT DATE September 1995		3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE THE DESIGN AND IMPLEMENTATION OF AN AUTOMATED INTELLIGENCE COLLECTION MANAGEMENT TOOL			5. FUNDING NUMBERS				
6. AUTHOR(S) Tomlin, Karen Denyse Hensley							
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER				
9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/ MONITORING AGENCY REPORT NUMBER				
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.							
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE			
13. ABSTRACT (Maximum 200 words) Currently, the only support for tactical intelligence collection management to a U. S. Army field commander is a manual process. This process results in a product that is many times erroneous and untimely. In response to this, the problem addressed by this work is to design and implement an automated collection management tool which will enable an intelligence analyst to provide a more timely and accurate intelligence picture of the battlefield.							
The approach taken was to design the tool using an Object-Oriented paradigm, develop the asset resource evaluation algorithms, and then implement the tool using U.S. Army Intelligence Community standards for interface design, coding, and functionality.							
This tool allows an intelligence analyst to translate a Commander's guidance into intelligence indicators composed of nodes with observable signatures, track collection assets, and evaluate the collection capability of assets against observable signatures by availability, capability, and vulnerability. The results of this thesis is an automated collection management tool which is presently under evaluation by the U.S. Army Intelligence Center and School for fielding at all echelons.							
14. SUBJECT TERMS Intelligence & Electronic Warfare, Collection Management, Requirements Management, Mission Management					15. NUMBER OF PAGES 110		
					16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified		20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited

**THE DESIGN AND IMPLEMENTATION OF AN AUTOMATED
INTELLIGENCE COLLECTION MANAGEMENT TOOL**

Karen Denyse Hensley Tomlin
Captain, United States Army
B.S., Oklahoma State University, 1984

Submitted in partial fulfillment of the
requirements for the degree of

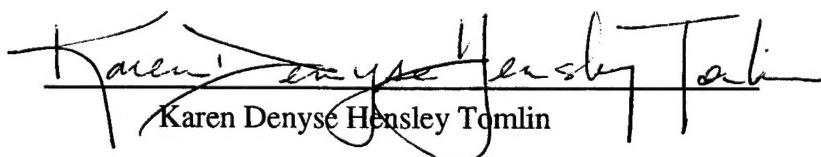
MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

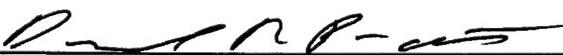
NAVAL POSTGRADUATE SCHOOL

September 1995

Author:

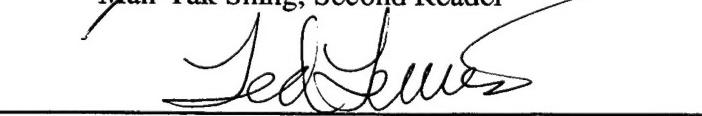

Karen Denyse Hensley Tomlin

Approved by:



David R. Pratt, Thesis Advisor


Man-Tak Shing, Second Reader


Ted Lewis, Chairman,
Department of Computer Science

ABSTRACT

Currently, the only support for tactical intelligence collection management to a U. S. Army field commander is a manual process. This process results in a product that is many times erroneous and untimely. In response to this, the problem addressed by this work is to design and implement an automated collection management tool which will enable an intelligence analyst to provide a more timely and accurate intelligence picture of the battlefield.

The approach taken was to design the tool using an Object-Oriented paradigm, develop the asset resource evaluation algorithms, and then implement the tool using U.S. Army Intelligence Community standards for interface design, coding, and functionality.

This tool allows an intelligence analyst to translate a Commander's guidance into intelligence indicators composed of nodes with observable signatures, track collection assets, and evaluate the collection capability of assets against observable signatures by availability, capability, and vulnerability. The results of this thesis is an automated collection management tool which is presently under evaluation by the U.S. Army Intelligence Center and School for fielding at all echelons.

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	BACKGROUND	1
B.	MOTIVATION	1
C.	PURPOSE	2
D.	METHODOLOGY	2
E.	ORGANIZATION	2
II.	PREVIOUS WORK	5
A.	ALL-SOURCE ANALYSIS SYSTEM-WARRIOR	5
B.	BATTLE COMMAND DECISION SUPPORT SYSTEM	6
C.	OPERATIONAL SYNCHRONIZATION MATRIX	6
D.	INTELLIGENCE & ELECTRONIC WARFARE SYNCHRONIZATION MATRIX	7
E.	RECONNAISSANCE & SURVEILLANCE SYNCHRONIZATION MATRIX	7
F.	SUMMARY	8
III.	OBJECT ORIENTED PROGRAMMING	9
A.	BASIC DEFINITIONS	9
B.	FEATURES	10
C.	BENEFITS	11
D.	APPLICATION INTERFACE ENGINE LANGUAGE	12
E.	SUMMARY	20
IV.	BACKGROUND	21
A.	OPERATIONS AND TACTICAL DECISIONMAKING	21
B.	INTELLIGENCE & ELECTRONIC WARFARE SUPPORT TO THE COM- MANDER	22
C.	INTELLIGENCE PREPARATION OF THE BATTLEFIELD	23
D.	COLLECTION MANAGEMENT	26
V.	DEVELOPMENT OF THE REQUIREMENTS AND MISSION MANAGEMENT MODULES	33
A.	DESIGN CONSIDERATIONS	33
B.	OBJECTIVE	35
C.	WHO ARE THE USERS?	36
D.	SYSTEM ARCHITECTURE	36
E.	IEW SYNCHRONIZATION MATRIX CODE DESCRIPTION	37
F.	DESIGN OF THE IEW SYNCHRONIZATION ANALYTIC ENGINE	44
G.	DESIGN OF THE IEW SYNCHRONIZATION MATRIX GUI	47
H.	SUMMARY	50
VI.	CONCLUSIONS	51
A.	INTRODUCTION	51
B.	EVALUATION	51
C.	FUTURE WORK AREAS	52

D. SUMMATION	54
APPENDIX A. SYNCHRONIZATION MATRIX USER'S GUIDE	57
A. STARTING AND EXITING THE APPLICATION	57
B. WORKING IN THE ASSET MANAGEMENT MODULE	59
C. WORKING IN THE REQUIREMENTS MANAGEMENT MODULE	64
APPENDIX B. WEATHER FORECAST INFORMATION	95
LIST OF REFERENCES	99
INITIAL DISTRIBUTION LIST	101

I. INTRODUCTION

A. BACKGROUND

Warfare in the 20th century is rapidly changing, both from a tactical standpoint as well as Operations Other Than War (OOTW). Advances in weapons technology have transformed the battlefield into a dynamic environment, one in which a commander must have timely, accurate intelligence in order to direct the force. It is critical that a commander take an active role in directing, integrating, and synchronizing intelligence information on to today's battlefield. Automating the information and decision making process are key to a commander's ability to maintain an advantage against a threat.

B. MOTIVATION

This research has been motivated by the need to automate the Collection Management process for the intelligence analyst so that timely and synchronized intelligence can be provided to the Commander in the tactical decision making process. Currently, the collection management process is manual and takes too much time compared to the fast-paced events on the battlefield. In order to support the tactical commander, intelligence analysts must be able to conduct Intelligence Preparation of the Battlefield (IPB) and provide an accurate situational 'picture' of the battlefield in a timely manner. By automating the collection management process, the commander will be provided a battlefield situational assessment quickly and allow greater decisionmaking and reaction time to enemy actions and maintain a tactical advantage. Additionally, the intelligence analyst will be able to monitor, evaluate, and alter the collection effort in real time in order to meet the dynamic challenges on the battlefield.

C. PURPOSE

The purpose of this research is to automate the Requirements and Mission Management portion of the Collection Management process by designing and implementing an automated tool and to integrate this software into the Intelligence and Electronic Warfare Synchronization Matrix prototype system. The Intelligence and Electronic Warfare Synchronization Matrix tool is a prototype intelligence planning system that can be used by an intelligence analyst at any echelon of the Army.

D. METHODOLOGY

To accomplish the successful implementation of the Requirements and Mission Management tool the following methodology was used:

- Analyze the role and importance that intelligence plays in the tactical decision making process.
- Understand the collection management process in intelligence preparation of the battlefield and the intelligence cycle.
- Outline the automation support requirements and capabilities required for both the system designer and the user.
- Design an automated tool which supports the information requirements of collection management.
- Implement the Requirements and Mission Management module of the Intelligence and Electronic Warfare Synchronization Matrix.
- Detail and describe the future of the Intelligence and Electronic Warfare Synchronization Matrix prototype and further enhancements and features that can be implemented to better support the user.

E. ORGANIZATION

This thesis is divided into six chapters. Chapter I is a general introduction of the thesis purpose, motivation, and background issues. Chapter II discusses current prototype development and gives a general overview of the computer language used. Chapter III covers the design and implementation of the Intelligence and Electronic Warfare Synchronization Matrix. Chapter IV discusses hardware and software issues, network communications, and database management. Chapter V outlines future work and

applications in automating the intelligence collection and synchronizing process. Conclusions, advantages and disadvantages are discussed in Chapter VI. The appendices include an explanation of the tactical and technical terms used within this document and a users guide that describes how the automated Intelligence and Electronic Warfare Synchronization Matrix works.

II. PREVIOUS WORK

The U.S. Army is now realizing that in order to maintain a tactical advantage on the battlefield, it must provide its soldiers advanced technology equipment and tools to do the job. This chapter provides a brief overview of two systems that are currently fielded that aid a commander in tactical battlefield management. The latter is an enhancement of the former. The remainder of the chapter describes three rapid prototyping projects in development prior to this thesis work.

A. ALL-SOURCE ANALYSIS SYSTEM-WARRIOR

The All-Source Analysis System-Warrior (Reconfigurable) (ASAS-W) is an all-source processing and data fusion system that supports automatic intelligence analysis, production, dissemination, and asset management. ASAS fuses threat information from all intelligence disciplines and provides correlated intelligence to maneuver commanders and staffs down to battalion level. Commanders use ASAS products to better comprehend enemy capabilities and intentions. At national and tactical levels, ASAS receives and correlates data from national, theater, and tactical intelligence sensors/sources and correlates the information to produce a common picture of the ground situation. ASAS assists intelligence managers to rapidly disseminate intelligence information; nominate targets, and manage Intelligence & Electronic Warfare (IEW) assets. This software system has mapping tools, task organization tools, enemy order of battle database tools, Operations Order (OPORD) tools, status reporting tools, course of action decision matrix tool, wargaming tools, and interactive graphics and networking interface capabilities. ASAS is the Army's premier intelligence analysis system designed to enhance the friendly force's lethality, survivability, and tempo on the battlefield. [ASAS95]

B. BATTLE COMMAND DECISION SUPPORT SYSTEM

The Battle Command Decision Support System (BCDSS) is an automated tactical planning and decision-making tool which allows the commander the ability to exercise battle command. The BCDSS baseline was developed from the ASAS-W workstation and the software has been developed by MYSTEC with the objectives of developing the tactical requirements for a common battlefield picture and to integrate the battle command into one display. The major capabilities that are integrated into the system:

- Interactive graphics
- 3D visualization
- Dynamic distributive overlays
- Voice activation
- Common scalable maps display
- Enemy and friendly force tracking
- Course of action analysis tool
- Operations Order (OPORD) generation tool
- Synchronization Matrix
- Video Teleconferencing

The system has been fielded on an experimental basis to various rapid deployment forces of the United States Army and has been successfully demonstrated to be interoperable with many of intelligence, signal, and field artillery systems currently deployed in the field. The BCDSS has proven to be a valuable tool to commanders and their staff and is scheduled to be used as a baseline, along with the Common Ground Station (CGS), for the Battle Command Decision Support System-Enhanced. [BCBL94]

C. OPERATIONAL SYNCHRONIZATION MATRIX

The Operational Synchronization Matrix (OSM) is a tactical planning tool that depicts each of the Battlefield Operating Systems (BOS), enemy actions, and critical decision points in a time and space relation. A feature of this Battle Command Support System is the ability to track each of the Battlefield Operating Systems: maneuver, fire support, air defense, intelligence and electronic warfare, command and control (C^2), engineer, and

sustainment, simultaneously in an event and time driven matrix. The matrix allows staffs to synchronize events on the battlefield during the wargaming process so that the best Course of Action may be chosen by a Commander. The OSM prototype has been designed and implemented by SAIC using a new software tool called Application Interface Engine. The Operational Synchronization Matrix software has been incorporated into the BCDSS using the available mapping data and database. The OSM software can also be used as a stand-alone tool for those units that do not have BCDSS.

D. INTELLIGENCE & ELECTRONIC WARFARE SYNCHRONIZATION MATRIX

The Intelligence & Electronic Warfare (IEW) Synchronization Matrix is a prototype intelligence analyst's tool that automates the asset management portion of the intelligence collection management process. The IEW Synchronization Matrix is currently a scalable design for use by units from Brigade through Echelon Above Corps (EAC) level. The current capabilities allow the analyst to task, schedule, and manage collection assets deployed on the battlefield. The IEW Synchronization Matrix depicts collection assets in a time driven matrix format which gives a visual depiction of which assets are available for scheduling and tasking during a predetermined period of time. The IEW Synchronization Matrix prototype has also been designed and implemented by SAIC using a new software tool called Application Interface Engine. It is to this matrix that this thesis project will be incorporated and enhance to a level in which the entire collection management process is automated.

E. RECONNAISSANCE & SURVEILLANCE SYNCHRONIZATION MATRIX

The Reconnaissance & Surveillance (R&S) Synchronization Matrix is a prototype intelligence analyst's tool that is similar to the IEW Synchronization Matrix but developed for units at Brigade and below. In addition to the capabilities of the larger scaled version, the intelligence analyst is also given the ability to develop a Commander's guidance into a collectable signature and then graphically depict asset taskings against the commander's

guidance. The R&S Synchronization Matrix also has a mapping capability which reads in Defense Mapping Agency (DMA) data from an active map server. The same government contractor and software tool was used to develop the R&S Synchronization Matrix.

F. SUMMARY

The ASAS and BCDSS are two systems that prove that the U.S. Army has realized the need to have an automated intelligence analysis system. Both systems provide a commander automated tools in which to manage the battlefield situation from an intelligence perspective. The three prototypes described provide a true analytical capability by evaluating a commander's requirements. The OSM provides a top level view of enemy COA's and friendly BOS's in a visual manner in order to show system synchronization. The IEW synchronization matrix, which only has asset management capability, and the R&S synchronization matrix provide the intelligence analyst a management tool to track requirements and assets. All three prototypes have been built into both ASAS and BCDSS and provide the requirements and asset management gaps that these two systems lack.

III. OBJECT ORIENTED PROGRAMMING

Style of design is an important consideration in the translation from concept to computer code. Although hardware can be an issue by restrict possibilities, the range of design possibilities remains broad. In order to avoid poor and incompatible designs, certain design models are used, to include Object Oriented Programming. [FUJA94]

Object Oriented (OO) Programming emphasizes the objects which operations act upon in contrast to the traditional programming method of emphasizing the algorithms and the order necessary to execute them. In this light, the OO paradigm is characterized by its emphasis on modules of code based upon items which can be considered ‘active’ data and these data values are closely tied to some actions and separated from others. Essentially, the data elements are ‘active’ in that they are both values and actions together. The object-oriented style represents an evolution of the structured programming models that most general-use programming languages are based upon. [FUJA94]

A. BASIC DEFINITIONS

In the OO paradigm, a group of data and actions is termed on object. The data values of an object are called attributes and the actions of an object are called methods. Objects perform their actions when they receive a message. The message causes objects to execute one or more methods, to send other messages, and ultimately to return a value. The return value is usually the object itself or another object that has been created. Objects are created and destroyed dynamically during execution of an object-oriented program.

Objects are instances of classes. A class defines the attributes and methods that its instances will have. When an object is created during processing, a copy of this class ‘template’ is made, various attributes are given values, and (in some cases) a new or initialization method is executed. many object-oriented languages organize their classes in an inheritance hierarchy. In this hierarchy, a class that is the subclass of another class

inherits all the attributes and methods of the superclass that the subclass does not purposely redefine. This system allows more specifically defined classes to be created from generic descriptions. Redefinition of attributes and methods occurs only when specifically required. [FUJA94]

B. FEATURES

Object-oriented languages normally have the following four main features:

1. Encapsulation

The concept of encapsulation is the ability to distinguish an objects' internal behavior and state from an object's external behavior and state. An object's data elements are hidden and therefore protected from manipulation by other objects within a program. The interaction of an object's data values are only available through message passing. Implementation of data and procedures within the object is not important to the program elements that only make use of the message interface.

2. Dynamism

The dynamism characteristic of an object gives it the capability to be created when needed or destroyed when no longer needed in a dynamic manner. The object in effect is managing its own existence. This feature is an advantage for memory management issues.

3. Inheritance

Inheritance is a mechanism by which objects have the ability to create new types by importing or reusing the description of existing types. The result is the organization of classes into a structured hierarchy that "... allows for increasing levels of specification, conservation of description for each class, and relationships among similar objects". [FUJA94]

4. Polymorphism

The term polymorphism is characterized by the capability of an object to take on many forms. This is also true in OO programming. When messages are sent to objects, within an inheritance hierarchy, to execute on its attributes, the class of the object must be evaluated to determine which method within the structure will execute. Because of the inheritance mechanism, an object belongs to a class for which it is an instance and also to each of its superclasses of that class. Besides executing their own specified methods, objects can also execute those methods from their superclasses.

C. BENEFITS

The object-oriented representation provides some important benefits in terms of security, maintenance, modification, conceptualizing, reusability, and policy implementation. Examples of the benefits accrued in these areas are as follows:

1. Security

Object data elements can only be modified by their own methods. In this way, they are preserved from corruption. In addition, the message interface ensures that other objects and actions are not affected by changes in the internal implementation of a given object.

2. Maintenance

The implementation of many actions into objects allows for improved traceability. Actions taken against a particular object can be easily detected, and the state of the object can be easily observed. The system is carefully divided into general activities and data structure manipulation activities; this division allows for easier localization of errors and more standard handling of errors.

3. Modification

Because of the abstraction provided by the objects, new programmers need not learn the implementation details of the objects in order to use them. Also, the connection

of all manipulative code with the object permits modification of the data structure and identification of all code that makes use of it.

4. Conceptualizing

The program is written in a way that more closely tied to its original concept of design. Physical objects can be mapped to application specific objects with characteristics defined as attributes and actions described as methods. This design methodology makes the program easier to understand, describe and document.

5. Reusability

Each object can be separated from the program for which it was developed and used in other programs with minimal effort. Reusing code by implementing application specific objects into other designs, prototype applications can be developed more quickly with tested code which can be bug resistant.

6. Policy

Using traditional programming methods, it is often difficult to implement the rules of the software application that operate at run-time. With the object-oriented framework, operational rules are easy to program. [FUJA94]

D. APPLICATION INTERFACE ENGINE LANGUAGE

1. General

The Application Interface Engine (AIE) is a state-of-the-art software tool developed by the Environmental Assessment and Information and Sciences Division of Argonne National Laboratory. It is a flexible development environment that facilitates information integration and management through graphically oriented computer applications [FUJA94]. Some of the features of this language are the graphical user interface, which enhances the program interface to its users; object orientation, which is increasingly becoming the standard technique for software development; modularity,

which facilitates portability and containability; distributed processing, and hardware standardizations. These features enable the software designer to concentrate on the design on the problem solution rather than the intricacies of code implementation. The AIE was chosen as the software development environment because of its suitability for rapid prototyping, consistency of graphical display and data access, its portability to various AIE platforms, its reusability of libraries of objects, and its adherence to the Army Common Operating Environment (ACOE) Standards.

2. Run-Time System

The AIE language provides a library of facilities for programmers. Unique to AIE, all of the facilities are implemented in the AIE run-time system library. Because AIE is object-oriented, the AIE library is composed exclusively of object classes. Each class has its own attributes and methods and because each class is a member of a class hierarchy, methods and attributes are also inherited from super classes. The objects that are available for use for the programmer are listed in Figure 1 and the explanation of the built-in attributes and methods can be obtained from the AIE Language and Object Reference Manual. [FUJA94]

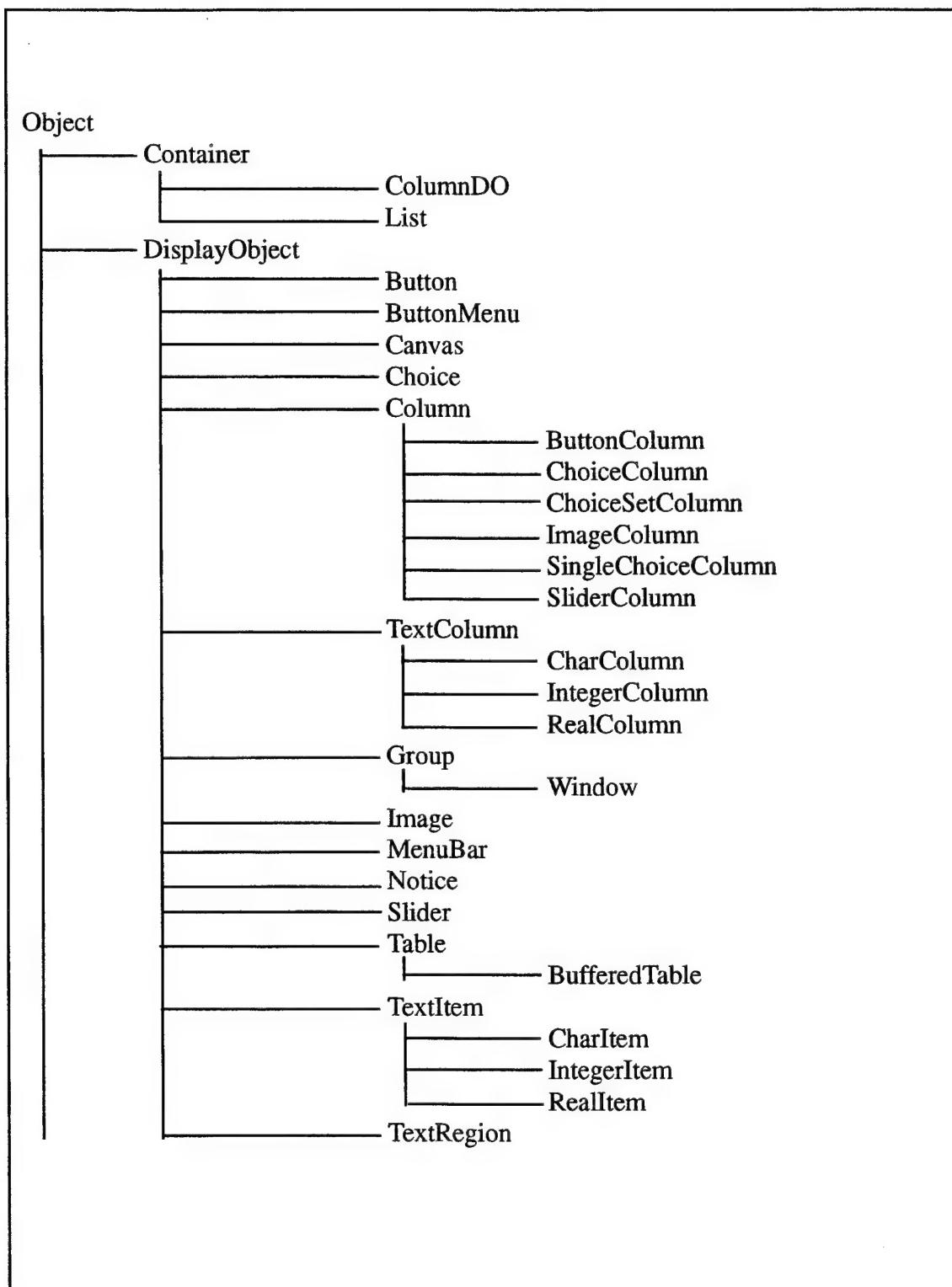


Figure 1: AIE Class Hierarchy

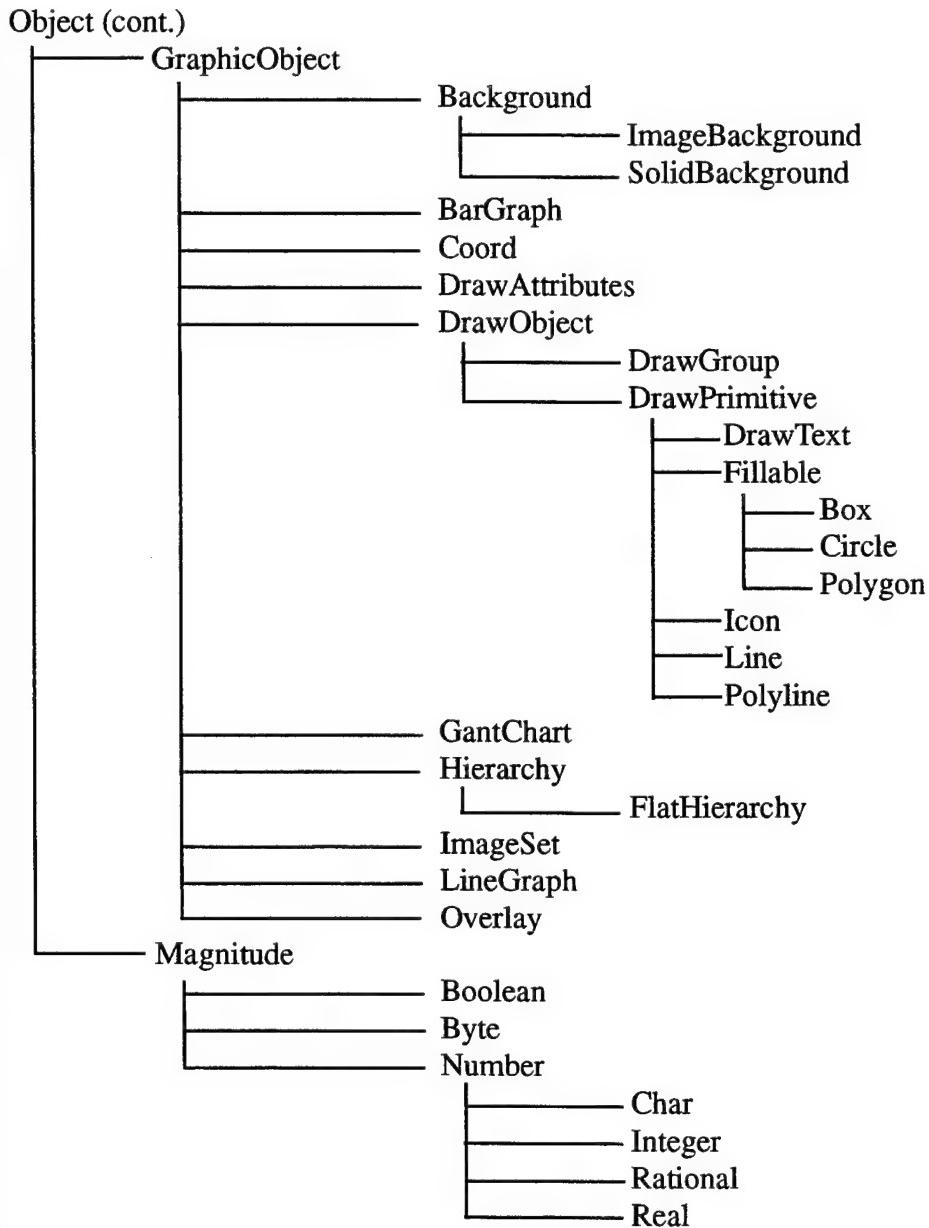


Figure 1: AIE Class Hierarchy

3. Compilation and Operation

An AIEL program consists of one or more units of translation contained in one or more system files. Each file is submitted to the AIEL compiler for translation into native system code. While program files can contain any number of units, two files will be produced for each file encountered during a compiler run: a native code file and a symbol dependency file. All files that are to be members of a single executable are processed by the AIEL linker to balance symbol dependencies. The linker creates a final native code file. Next, all native code files are submitted to the native code compiler and system loader to produce the desired executable. AIEL uses C as its native code programming language. Because C is available on many platforms, AIEL is made directly portable to these platforms. [FUJA94]

After a program has been parsed and analyzed, code generation occurs. The code generated by the AIEL compiler is written in the native system language. This code is then compiled by the native compiler and linked into the run-time library and the toolbox libraries. Figure 2 shows how AIEL code is converted into executable code.

The operation of the AIEL system follows a set of rules. Programmers who decide to use AIEL as a software design environment should have a broad understanding of the concepts of inheritance, containment, scoping, message resolution, typing, and initialization.

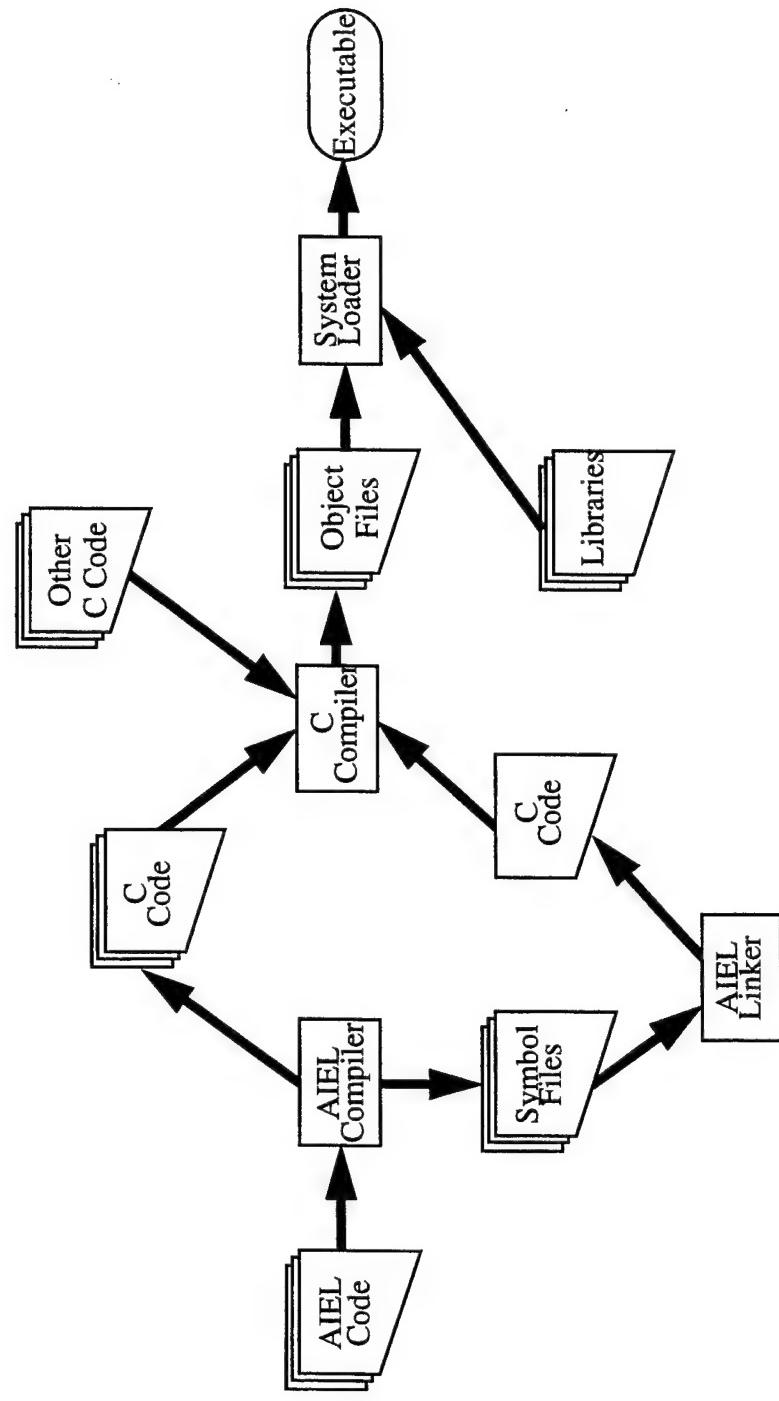


Figure 2: AIE Compilation

a. Inheritance

Specifically with the AIEL, the inheritance mechanism ensures classes are arranged in a strict relationship with other classes. Each class has a position in the hierarchy. When a class is created, its parent class, also called superclass, is specified. The new class inherits all the attributes and methods from its superclass, but is distinctly different due to the new class' own attributes and methods. If a new class redefines an attribute or method which has the same name as its parent, the superclass' defined attributes and methods are completely hidden (with exception of the super keyword for methods).

b. Containment

The containment concept refers to the relationship between objects and other object's attributes. In effect, objects contain their attributes. In AIEL, containment is taken care of by dereferencing and through the parent keyword. Not all attributes have parent pointers. Attributes that are simple value types (ex. Integers) do receive parent pointers. In other cases, such as Lists or Display Objects, objects are provided with a pointer to its parent and the parent keyword refers to this pointer. Currently, variable resolution follows the containment mechanism when a variable reference cannot be solved within the current object.

c. Reference Resolution

The scoping mechanism used by AIEL is such that when an identifier refers to another object, it can take one of two forms - a single-word indirect reference or a dot-path direct reference. The dot-path method can cause object security problems and therefore the preferred method is by using AIEL's built-in reference resolution mechanism. AIEL uses a strict multi-step methodology for determining which object, local variable, or parameter is referred to and will notify the user if no object corresponding to an identifier exists. The sequence of seven resolution attempts is as follows:

- If within a method and a matching variable local to the method exists, it is produced.

- If the reference matches a formal parameter of the current method, the parameter is produced.
- Attributes of the object executing the method (if in a method) or other attributes of the current object (if in an attribute definition) are checked for matching names. This includes attributes inherited from superclasses. If a matching attribute is found, its reference is produced.
- Attributes of the object that contains the current object are checked.
- Attributes of the parent of that object are checked recursively until the unit level of containment is reached.
- If no object corresponding to the identifier is found upon using this sequence, an error message is printed and an optional debugging core is produced. [FUJA94]

d. Message Resolution

When a message is sent to an object, the resolution of the activated method is decided upon by a strict methodology. Starting at the class definition for the object to which a message is sent, a method searches for an exact name and parameter arity that match the message. If a method cannot be found at that level, then the method will traverse up the hierarchy to the next class level for resolution. If no match is found, a system error message will be displayed.

e. Typing

The variables and formal parameters in AIEL are not typed. A variable can be reset to another type merely by assignment. Likewise, formal parameters can also be of any type. When a method is invoked, its name and parameter arity only are checked, this may seem like it would bring more confusion to not having type checking, but in fact using this system produces more robust methods that can react to different parameter types. Although the variables and parameters in AIEL are not typed, the objects in AIEL are typed. All objects belong to a particular class and its class is similar to its type.

f. Initialization

In AIEL, the initialization order and other dependencies are unique. When a program is compiled, a series of C structures corresponding to the class descriptions is produced, along with a number of other procedures that initialize the attributes. Each

attribute is initialized in the order in which it appears in the AIEL file. Each object is fully initialized prior to the next object in the file. When an object is initialized, its attributes are initialized first, and then the object itself is produced. This is a recursive process. Because of the manner of initialization, AIEL allows for backward referencing only.

4. Hardware & Software Requirements

a. Hardware

The AIE is available and can be used on Sun Workstations and on IBM personal computers and compatibles. As a result of AIE's modular design, the same AIEL applications can be compiled on both platforms.

b. Software

The AIE utilizes and meets the standards for X Windows, Motif, Microsoft Windows, Structured Query Language (SQL), and the Transmission Control Protocol/Internet Protocol (TCP/IP). Additionally, a Linux version of the AIE is under development.

E. SUMMARY

Object-oriented programming is a very popular and in some instances a very necessary method for solving problems, both in concept and code. For the purposes of this thesis, an object-oriented programming language was required for ease of implementation in resolving the problem.

AIEL was the most appropriate language to use because of its object oriented features and high level programming approach which made it much easier to program and allowed for more time in problem design and solution. Additionally, besides meeting DoD computer software standards, several of the prototype system discussed earlier; OSM, IEW Synchronization Matrix, and R&S Synchronization Matrix, were developed using AIEL and to optimize on the reusability aspect this language was chosen for this project.

IV. BACKGROUND

In developing the automated Intelligence and Electronic Warfare (IEW) Synchronization Matrix, there are a few concepts which are extremely important and must be understood. The decisionmaking process that a tactical commander must go through and the role that IEW plays in this effort are paramount. By conducting Intelligence Preparation of the Battlefield (IPB) and then going through the Collection Management (CM) process, intelligence synchronization can be realized which can greatly influence how a commander orchestrates the battle. This can give a commander the tactical edge and afford him the capability to make smart decisions and direct the force.

A. OPERATIONS AND TACTICAL DECISIONMAKING

Tactical decisionmaking is a very dynamic multidimensional process that must allow decisions about current operations to occur simultaneously with decisions and planning about future operations. This process is a continuous cycle that flows from information to planning to decisions to execution and repeats throughout the process. A commander and his staff must work in concert in order effectively evaluate the battlefield and reach decisions. A commander is responsible for making decisions. His staff helps in this process by providing timely and accurate information and executing the decisions.

The decision making process is a four step process as follows:

- Mission Analysis - The first step in tactical decision making. This phase involves gathering facts, making assumptions, analyzing higher headquarter's mission and intent, and issuing commander's guidance.
- Course of Action Development - The second step in the tactical decision making process. This phase consists of analysis of relative force ratios, array initial forces, development of a scheme of maneuver, determining command and control (C2) means and maneuver control measures, and preparing courses of action statements and diagrams.
- Analysis of Courses of Action - The third step in the tactical decision making process. This phase is involved in war gaming each of the proposed courses of action

to determine the best possible Courses of Action (COA) to recommend to the commander for implementation.

- Decision and Execution - The fourth and final step in the tactical decision making process. In this phase, the commander announces his decision and the staff makes the necessary preparation for execution. Upon implementation of a plan, the commander and staff trigger the decision making cycle in motion by monitoring the current operations and making changes to their plans as the situation requires.

Doctrinally, each staff officer has a specific role in supporting the commander throughout the decision making process in choosing a course of action. The intelligence staff officer is responsible for providing the possible enemy courses of action as well as orchestrating the friendly intelligence collection effort all in support of the mission.

B. INTELLIGENCE & ELECTRONIC WARFARE SUPPORT TO THE COMMANDER

Intelligence and Electronic Warfare (IEW) operations are key to a commander's victory in war and success in operations other than war. Commanders use IEW to focus the combat power at their disposal to win decisively. Commanders also use IEW to protect and conserve power and resources during operations. [FM34-1]

Intelligence and electronic warfare support to the tactical commander is carried out by the Intelligence Officer (G2 or S2, depending upon echelon) who must be trained to understand the dynamics of combined arms operations and how to synchronize the intelligence effort with the commander's concepts. [FM34-1] This is not only true in the war time situations, but in the new 'Force Projection' Army, IEW operations are a key and fundamental aspect to successful Operations Other Than War (OOTW). IEW support to the force projection Army is based upon five principles: the commander drives intelligence, split-based operations, tactical tailoring, broadcast dissemination, and what this thesis is based upon, intelligence synchronization. The commander's role in IEW operations begins well before a conflict begins and continues throughout the operation. He is responsible for directing the intelligence collection effort by driving the prioritizing of intelligence and targeting requirements. Split-based operations provide the commander continuous, timely,

and accurate IEW support during wartime operations through tailored organizations with access to national level collection assets. The commander tactically tailors units to provide IEW support based upon the mission, the contingency operations, and the intelligence requirements which allows a more efficient and effective support unit. Broadcast dissemination of intelligence and targeting information provides commanders at each echelon the ability to ‘see the battlefield’ in a common view. Intelligence synchronization ensures that IEW operations are linked to the commander’s requirements and responded to in time to influence decisions and operations. In the synchronization process, the intelligence analyst takes the commander’s priority intelligence requirements (PIR) and plans backwards to ensure that collection production efforts are orchestrated with the operation, and deliver intelligence when required. The collection manager ensures specific orders and requests (SORs) fully support all PIR and information requirements (IR). The collection manager also synchronizes collection and reporting to deliver relevant information, on time, to support operational decisions. Intelligence synchronization also ensures that the MI unit commander has the time, guidance, and resources to execute IEW operations. Intelligence synchronization is a continuous process which keeps the intelligence cycle and IEW operations tied to the commander’s critical decisions and concept of operations. [FM34-1]

C. INTELLIGENCE PREPARATION OF THE BATTLEFIELD

Intelligence Preparation of the Battlefield (IPB) is the process of analyzing the enemy situation and potential courses of action and terrain and weather effects on the battlefield and on fighting forces. IPB will provide the tactical commander with the information necessary in the tactical decisionmaking process of how to apply and maximize combat power on the battlefield. IPB is a continuous process that is conducted in four systematic steps:

1. Define the Battlefield Environment

This is the process of identifying characteristics of the battlefield that will effect both friendly and threat operations, determining the friendly area of interest (AI), identifying gaps in current intelligence holdings.

2. Describe the Battlefield's Effects

This is the process of identifying the effects that the battlefield environment has on the friendly and enemy forces.

3. Evaluate the Enemy

This is the process of determining the courses of action that the threat forces may take as a result of the effects of the battlefield environment and its effects.

4. Determine Threat Courses of Action

This is the process of identifying and developing likely enemy COA's that will effect the friendly mission.

Figure 3 illustrates these steps. Not only is IPB conducted during the tactical decision making process, but continues throughout the actual conduct of operations.

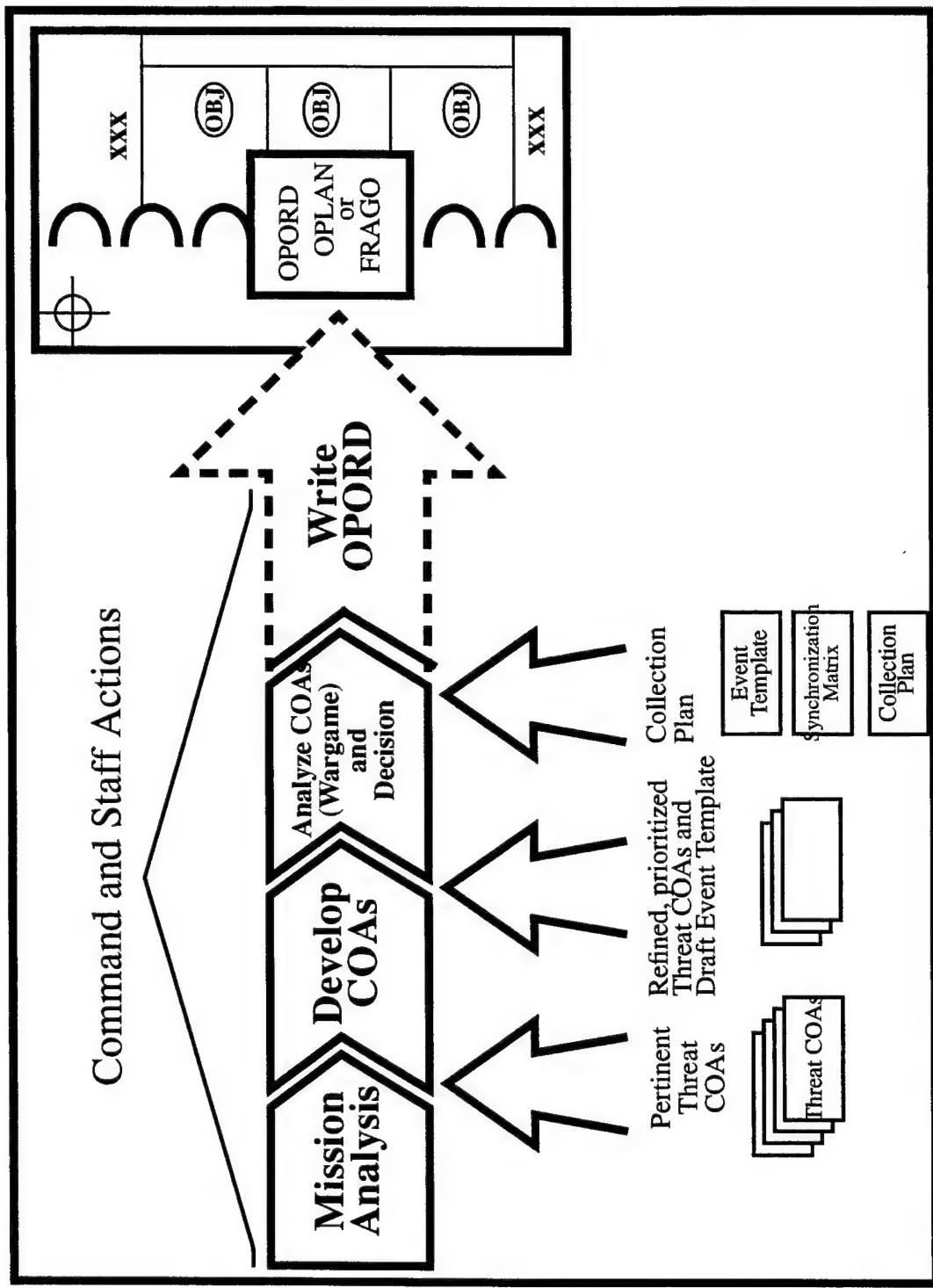


Figure 3: IEW supports the decision making process

The IPB and the tactical decisionmaking processes have an inherent relationship in that the IPB process is an essential element to and must be incorporated with each step of the tactical decisionmaking process. In the mission analysis phase, a commander uses the IPB products to assess current battlefield situation and to make assumptions as to the interactions between the friendly and threat forces. In the courses of action development phase, a commander's staff develops the friendly COAs based upon the mission analysis step and the IPB. In the COA analysis and comparison phase, the staff wargames the different threat COAs developed in the last step of the IPB process. In the decision phase when the commander has chosen a COA and has developed the appropriate PIR/IRs, the IPB products are again used by the intelligence officer to formulate a collection plan that will support and satisfy the commander's guidance. Finally, in the execution phase, the IPB process continues identifying new intelligence requirements and reevaluating the current situation.

The IPB process is the cornerstone of the intelligence effort during the wargaming phase of the tactical decisionmaking process. As a result of this process, facts and assumptions about the battlefield environment and threat are determined and the intelligence collection effort and synchronization with other battlefield operating systems is focused.

D. COLLECTION MANAGEMENT

Collection Management is a set of procedures that orchestrate the Intelligence Systems of Systems (ISOS) organizations and systems to focus the intelligence effort in support of warfighting and operations other than war. [FM34-2] CM provides the tactical commander the intelligence required to determine friendly COAs and targeting priorities. The collection management process includes three distinct subfunctions: Requirements Management (RM), Mission Management (MM), and Asset Management (AM). These sub-functions distinguish between internal and external relationships among collection

managers, requestors, and collectors during CM operations. Figure 4 shows these functional relationships. [FM34-2]

The collection management process consists of six steps:

- Develop requirements
- Develop collection plan
- Task or request allocation
- Disseminate
- Evaluate reporting
- Update collection planning

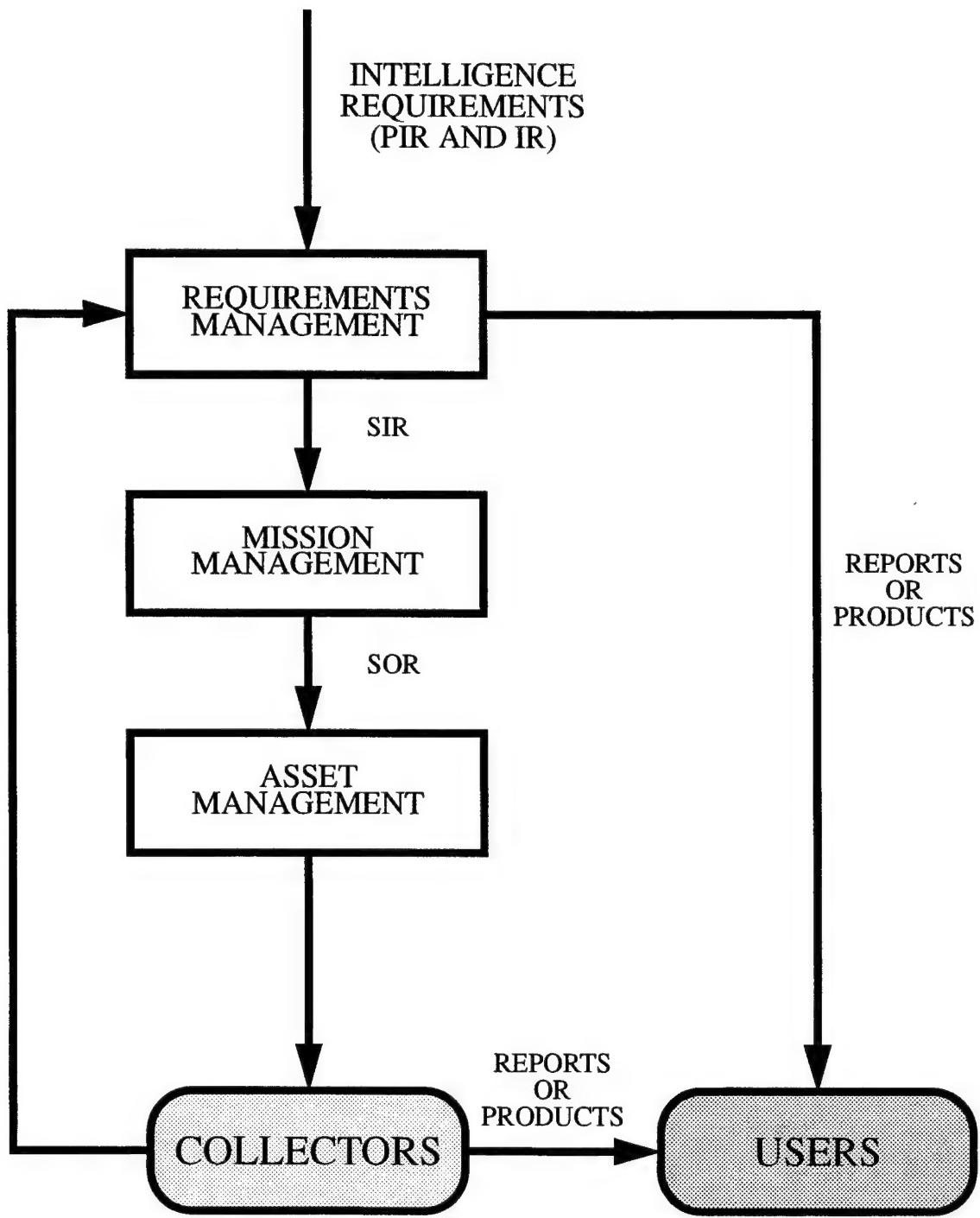


Figure 4: Collection Management Relationships

The first step involves identifying, prioritizing, and refining the uncertainty issues that pertain to the threat and the battlefield environment and must be resolved in order to accomplish the mission. The second step involves developing an integrated and synchronized plan that selects the most suitable collection assets for each of the intelligence requirements. The third step involves the actual implementation of the collection plan through the execution of system-specific tasking or requesting mechanisms. The fourth step involves ensuring product delivery to all subscribed customers in a timely manner. The fifth step involves the evaluation of how well the implemented collection plan is satisfying the commander's requirements. This step involves making necessary revisions to the overall collection plan in order to better fully synchronize and optimize the collection effort. Each of the six steps is the responsibility of one of three subfunctions, with some overlap. Figure 5 shows the steps and subfunction relationship overlaps.

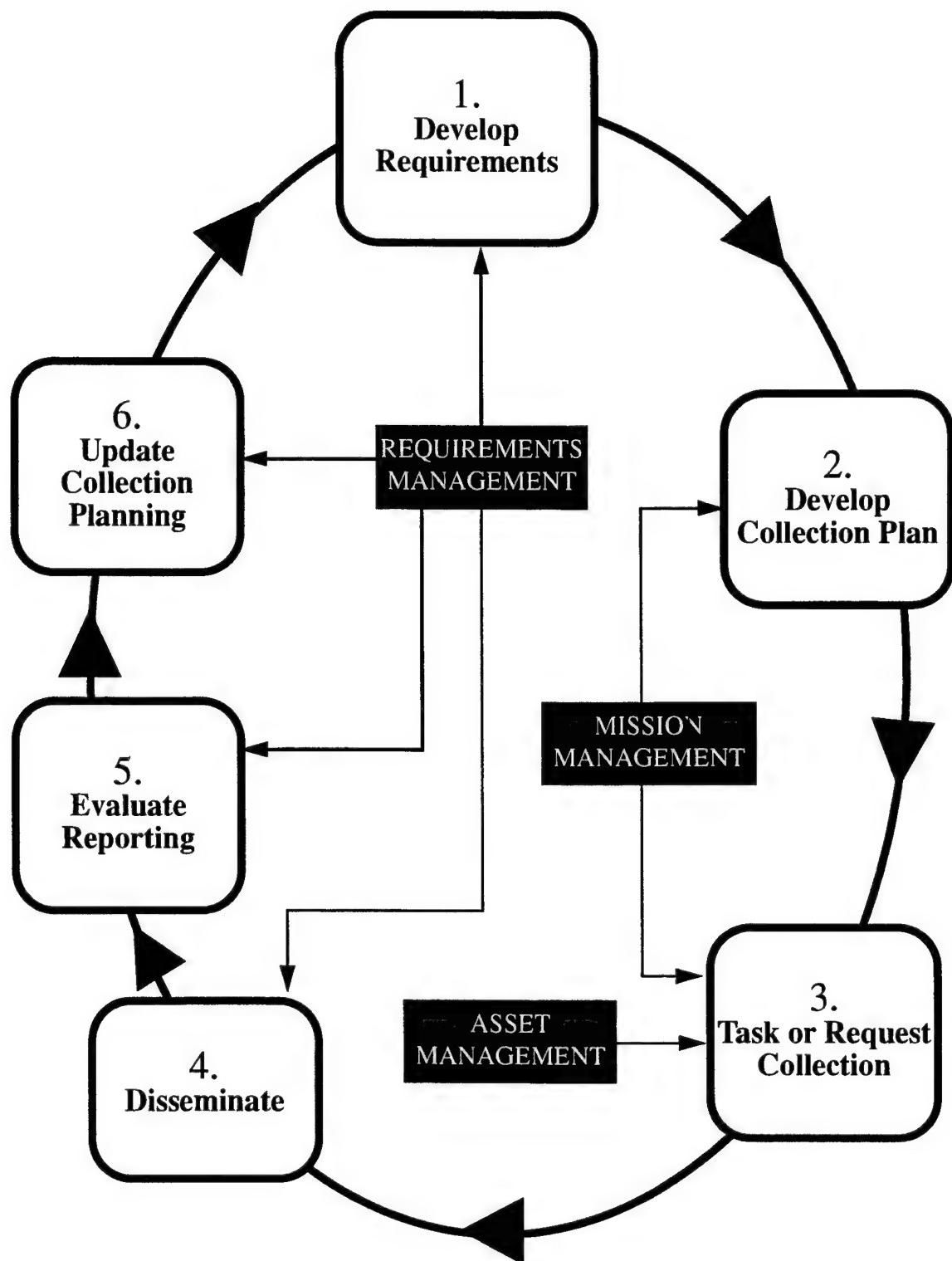


Figure 5: Collection Management Sub-Functions & Relationships

1. Requirements Management

Requirements Management (RM) is involved in the development of the requirements and collection plan, dissemination, evaluation of reporting, and updating of the collection plan steps of the collection management process. RM defines what to collect, when, and where. Once the commander's priority intelligence requirements (PIR) and information requirements (IR) are determined, requests for intelligence collection are developed (these can include requests from outside agencies). The collection manager reviews the intelligence requests for completeness, pertinence, and feasibility. Once the requests are validated, the requirements manager checks local databases to determine if any of the intelligence requests can be immediately satisfied. If not, a new requirement is created for collection. The requirements manager integrates new orders and requests for intelligence into the existing command's requirements list, re-prioritizes the list if necessary, and then develops the Special Intelligence Requirements (SIR).

Correlating intelligence reporting to the original requirement and evaluating the reports are key sub-functions in of RM. This is the quality control effort that helps ensure timely satisfaction of intelligence requirements. RM also includes dissemination of reporting and related information to original requestors and other users. [FM34-2]

2. Mission Management

Mission Management (MM) is involved in the development of the collection plan, the tasking or requesting allocations, and the updating of the collection plan steps of the collection plan. MM defines how to employ the intelligence collection resources to satisfy the mission requirements. The mission manager is responsible for evaluating the suitability of collection systems, units, and other agencies based upon the capability, availability, vulnerability, and performance history. The MM process maps out a collection strategy which involves synchronizing collection schedules with the PIRs and derives the specific orders and requests from the SIR. The collection strategy is revealed through the collection plan. MM generates the actual collection task and requests and continually

monitors collection asset status. Another responsibility of MM is exploitation management. Exploitation management uses intelligence processing equipment to make intelligence collected by theater or national agencies available to the tactical users. By incorporating exploitation management into the collection planning process, commitment of organic tactical systems can be better utilized. [FM34-2]

3. Asset Management

Asset Management (AM) is primarily involved in the tasking or requesting allocations step of the collection plan. AM executes collection and/or exploitation in accordance with the collection plan requirements. [FM34-2] AM integrates the RM process of what, when, and where to collect with the MM process of how to employ resources and executes the collection mission with specific assets and resources. The IPB and collection management processes also have a complementary relationship. While the IPB process aids a commander in identifying new intelligence requirements and providing the direction to satisfy them, the collection management process synchronizes the events of units and resources in order to provide timely intelligence information to the commander in support of the mission.

4. Summary

Every commander goes through the tactical decision making process in one form or another during the wargaming process. Each commander has their own method of conducting the procedure. In every case, however, a commander must rely upon the Intelligence Officer to provide as accurate as possible a picture of the battlefield and how the enemy will attempt to use it to his advantage and the best plan to utilize the available intelligence collection assets. This is done by IPB and CM processes. The collection management process and intelligence synchronization are critical components to a commander's ability to make timely decisions to influence the battle.

V. DEVELOPMENT OF THE REQUIREMENTS AND MISSION MANAGEMENT MODULES

There are several key issues affecting command, control, communications, computers and intelligence support system developments. The more prominent include:

- Affordability: Dominates system considerations as decreasing budgets force tough decisions on program developments.
- Interoperability: Essential at all levels. Within the Synchronization Matrices realm, the compatibility between design and network management reduce the need for costly, system-unique interfaces.
- Integration: Integration of new systems means intensification of management efforts to ensure all the pieces, including supporting communications and additional equipment, are fielded and integrated in a synchronized manner to each operational force.
- Software: Software design, development, and sustainment are the most expensive elements of automated systems and can easily exceed allowable budgets if not managed properly. New systems must be designed and tested within Army tactical Command and Control Systems (ATCCS) operating environment to ensure full integration and interoperability before full-scale development and production.
- Testing: Testing cannot be done in isolation, following traditional approaches.
- Training: Must start early, be continuous and focus on commander and staff involvement. [WAYNE90]

These key C⁴I issues were of paramount concern during the design portion of this thesis and where applicable the concepts were implemented to provide a most robust and acceptable product.

A. DESIGN CONSIDERATIONS

Staffs use wargaming to develop, refine, and compare possible friendly courses of action. This wargaming enables the staff to do the necessary comparisons in order to select the best COA for recommendation to the commander. The IEW Synch Matrix needs to be a flexible tool that allows an analyst to conduct the collection management process on each of the developed COAs.

The collection management process is virtually a uniform process that does not change with tactical echelon (battalion, brigade, division, corps, or echelon-above-corps) or operational mission (joint, combined, or interagency). The actual collection plan provides the structure for the development and evaluation of intelligence requirements. The ‘plan’ is then used to satisfy those requirements. Due to the diversity of missions, capabilities, and requirements, the collection plan has no prescribed doctrinal format. [FM34-2] Thus, the collection can be tailored to meet a commander’s needs. There are, however, characteristic features of a dynamic plan that should be taken into consideration:

- Have as its basis the commander’s intelligence requirements.
- Help the commander see as deep in depth and time as possible.
- Cover deep, close, and rear operations.
- Have a four dimensional battlefield approach: width, length, height, and time.
- Cover the collection capabilities of higher and adjacent units.
- Be flexible enough to allow response to changes as they occur.
- Cover only priority requirements.
- Be a working document.
- Contain precise and concise language. [ST100-9]

In this vein, one of the first design considerations was that this tool must accurately represent the collection management process that is currently conducted in the intelligence TOCSEs. The second design consideration was to determine the scope of the collection management process that this system would represent. There have been numerous discussions among both commanders and military intelligence analysts who have experience with the current method of operation and who will be going back out to units where this system will be fielded. Some of the questions that have arose and must be addressed are:

- Who are the users and what are their requirements?
- What will a Commander and Intelligence Officer expect from a collection process tool?
- How can an automated collection tool fulfill the expectations of its user?
- What kind of information will be required to provide to the users and where and from what source will the information be derived?

- How can the tool be designed so that it can be easily integrated into existing systems?

During the design of the automated Collection Management Tool there were several principles that were kept in mind:

- Specify the objectives.
- Identify the users, their roles, and the decisions they will be expected to make.
- Determine the information they will need to make the decisions, and identified the information feedback required to achieve the tool's objective.
- Design an interface to insure the efficient manipulation of intelligence data that will meet the needs and requirements of the users.
- Design a system to insure easy integration into existing prototype systems as well as the ability to act as a stand alone model.
- Consider the possible network architecture environment.
- Keep the design simple yet powerful.
- Test and evaluate the design in the field.

B. OBJECTIVE

Intelligence Tactical Operations Center Support Elements (TOCSE) are characteristic of having large maps with acetate overlays depicting the current enemy situation plan and friendly collection schedules that support the commanders plan. Many long hours of work by just as many analysts are put into developing enemy doctrinal, situational, and event templates. A commander and his intelligence staff rely upon these overlays to help plan and conduct the necessary wargaming in the friendly COA selection process. Once the enemy templates have been completed, the tactical commander becomes involved and thereafter dictates the changes to the overlays. This manual effort is both time consuming and quickly becoming a detriment to a commander's ability make quick intelligent decisions on how to direct the battle. The time and effort spent in the construction and maintenance of the enemy situational overlays could be drastically be reduced if the entire collection management process could be automated. Commanders and Intelligence Officers could better utilize their time analyzing the enemy situation and formulating possible courses of action. In order to speed up the process and the accuracy of the collection management cycle, automation of this process is a necessity.

The objective of this design is to automate the intelligence collection management effort into a simple, easy to use analyst's tool that will enable the intelligence analyst to manage the intelligence mission by tracking and tasking collection assets while providing a current enemy situation assessment at a moments notice.

C. WHO ARE THE USERS?

The users of the IEW Synchronization Matrix will be Intelligence analysts, both Officer and Enlisted, at each level of the operational and tactical echelons in support of a commander. The commander drives the mission and makes the decisions. The intelligence staff is responsible for providing the commander the necessary information to make the decisions and steer the battle. This tool can be used by both single and multiple users within an Intelligence Tactical Operations Center Support Element (TOCSE), and by multiple users both within an intelligence analytical cell and across a network at other echelons.

D. SYSTEM ARCHITECTURE

The system structure of the automated CM tool consists of the analytic engine, user interface, and the database. Figure 6 shows the relationship between the structure elements and the users.

Both the Graphical User Interface (GUI) and the analytical engine modules are designed using AIEL. The database used by the system is dependent upon the user's base system. Those units that have the ASAS-W system will have the synchronization matrix integrated and allow the matrix to retrieve, manipulate, and save data. Elements that do not have the ASAS-W system will use their own database which can be as simple as text files or using a structure database system.

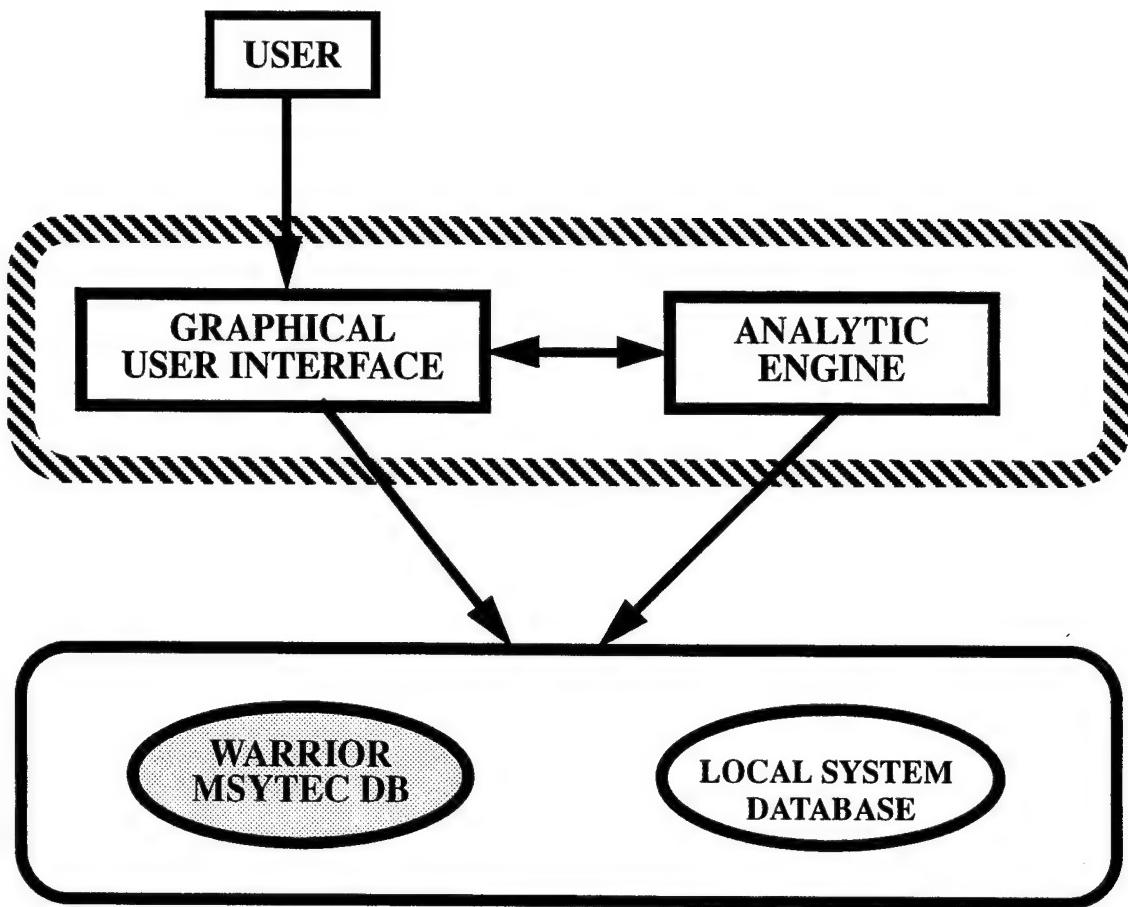


Figure 6: Collection Management System Architecture

E. IEW SYNCHRONIZATION MATRIX CODE DESCRIPTION

The collection management problem can be broken down into two main modules: collection assets and commander's priority intelligence requirements. Each of these modules has its own hierarchical or containment relationship structure that will be defined and explained in this section.

1. Collection Assets

The class hierarchies of a collection asset, generic and instantiated, are shown in Figure 7 shown below.

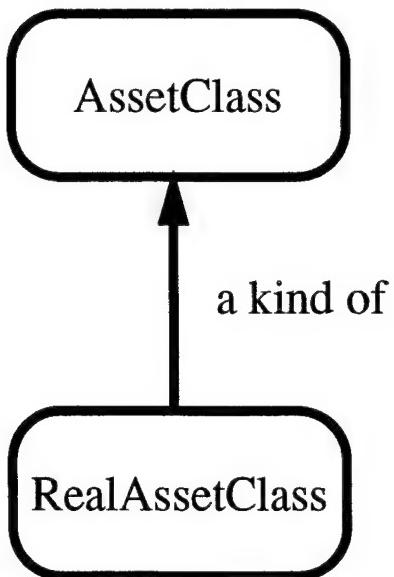


Figure 7: Collection Asset Object Hierarchy

The AssetClass is the superclass that represents a generic asset and contains the following attributes with explanations:

- InstanceName - Army/navy nomenclature of the system
- Caveats - SCI field
- Classification - 0=Unclass, 1=Confidential, 2=Secret, 3=Top Secret
- Compartments - NIL=None, 1=NATO, 2=NOFORN
- Owner - Echelon and/or unit that owns the asset
- PrettyName - Nickname for the system
- AssetType - Type of collector (COMINT, ELINT, IMINT, HUMINT)
- FrequencyMin - Minimum frequency (SIGINT only)
- FrequencyMax - Maximum frequency (SIGINT only)
- Modulation - Modulation types of the asset
- Standoff - The range in kilometers that the sensor must operate from
- TaskingLeadTime - Time in hours that the system must be tasked ahead of mission
- ProcessingTime - Time in hours that the system needs to process raw data into intelligence
- ReportingTime - Time in hours that supporting communications take to forward intelligence

- Power - Future field
- MTBF - Mean Time Between Failures for a system expressed in hours
- MTTR - Mean Time To Repair for a system expressed in hours
- MissionLoadMax - Maximum number of taskings which can be handled by the asset
- Endurance - Length of time in hours that the system can operate
- ReportType - JINTAAC message format produced by the asset
- CantCollect - Emitters that the asset is not capable of collecting against (SIGINT only)
- AdverseWeatherEffects - Types of weather which degrade the sensor performance
- EnemyThreatToAsset - Enemy systems which present the greatest threat to sensor
- ImintResolution - Number in meters
- Catalog - Type of asset (ground, air-breathing, non-air-breathing)
- DoesRangeEffect - Boolean value
- DoesLOSEffect - Boolean value
- Range - Normal range in kilometers that the sensor can normally detect
- PlatformData - Name used to point to the Platform-Database
- Remarks - Free text field

Each of the attributes are used by the analytical engine in determining whether a system is capable collecting against a PIR. Each of these attributes has a unique value for each type of generic asset in the database.

The RealAssetClass is a subclass of the AssetClass that represents an instantiated collection asset. This subclass inherits the attributes of the AssetClass, the values of a specific type of asset, plus the following:

- Id - Computer generated Id field
- EntityId - Id of a particular asset (bumper number, wing number)
- Location - Location of asset in UTM on the ground or center of track (aerial)
- Schedule - List of schedules available collection times
- Taskings - List of taskings
- AssetYPos - AIE definition of placement on lower canvas of GUI
- Up - Boolean value
- Capable - Boolean value

- InRange - Boolean value
- Timely - Boolean value
- LOS - Boolean value
- MaxTaskings - Boolean value
- EnemyThreat - Boolean value
- WeatherThreat - Boolean value
- Scheduled - Boolean value
- Status - Overall capability of asset to collect against a PIR (8-10-Capable, 5-7-Marginal, 0-4-Not Capable)

The last nine attributes (excluding Status) are boolean values that are locally set as a result of an asset being compared to a specific PIR. The final attribute, Status, is calculated as a result of the boolean valued attributes ‘True’ cardinality.

2. Commander’s Priority Intelligence Requirements

Priority Intelligence Requirements are developed from a commander’s operational guidance as to who, what, when, and where the enemy may conduct operations within the friendly area of operations. The PIRs are broken down into indicators. The indicators break requirements into smaller, more specific information requirements. Indicators are then transformed into Specific Intelligence Requirements (SIRs) by rephrasing the indicators into questions which, when answered, can satisfy the larger intelligence requirements. SIRs describe what information is required, where on the battlefield it can be obtained, and when it is to be answered. SIRs are as detailed as possible [FM34-2]. SIRs can be subdivided into nodes. Nodes are defined as generic units that depict the type and size of unit. Each node is broken down into characteristics signatures (COMINT, ELINT, and IMINT) which are physically collectable. Figure 8 depicts a PIR-Signature breakout notional example and Figure 9 depicts the PIR-SIR (Indicator)-Node-Signature object hierarchy.

PIR: When will the 15th Tank Division attack?

SIR: Activation and forward/lateral movement of regiment CP's.

NODE: Infantry Regimental Forward CP

IMINTSIG: Infantry Regiment Forward will occupy an area 50 m²

COMINTSIG: R-105, E-459, A7A, RBM-1

ELINTSIG: N/A

NODE: Corps Forward CP

IMINTSIG: Corps Forward CP will occupy an area 3x4 km will

consist of 5-8 V-415 Jeeps

COMINTSIG: KV-M, RSB-F

ELINTSIG: N/A

NODE: Armor Regiment Forward CP

IMINTSIG: Armor Regiment CP will occupy an area 1x2 km and
will consist of 1 T-54, 1 BTR-60, 1 V-415 Jeep, 10 2T Trucks, and
a Helipad.

COMINTSIG: R-109, R-106, 308, 9-RS, 12RP, RBM-1

ELINTSIG: N/A

SIR: Sustained artillery attacks without immediate follow-up ground attack.

NODE: Artillery Battery

IMINTSIG: Arty Bty will occupy an area 500 m² and consist of
1 V-415 Jeep

COMINTSIG: R-108, E-459, A7A, 12-RP

ELINTSIG: N/A

NODE: MRL Battalion

IMINTSIG: N/A

COMINTSIG: R-108, E-459, A7A, 12-RP

ELINTSIG: N/A

NODE: MRL Battery

IMINTSIG: N/A

COMINTSIG: R-108, E-459, A7A, 12-RP

ELINTSIG: N/A

Figure 8: PIR Hierarchical Text Example

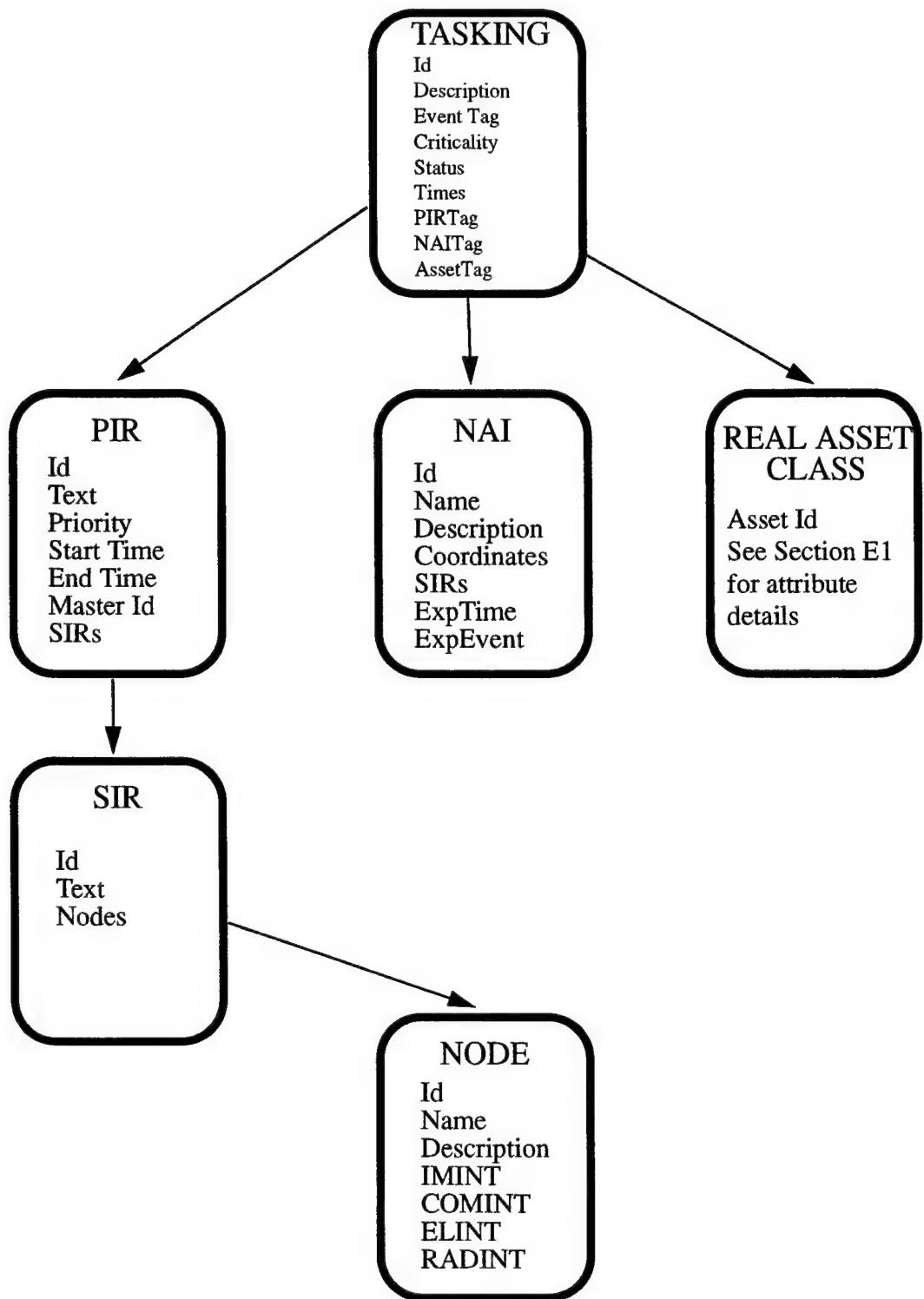


Figure 9: PIR Hierarchical Definition

Each level of PIR hierarchy is represented as an object and has a containment relationship with its higher level. The PIRClass object has the following attributes:

- Id - Combination of PIR and SIR priority
- Text - Text of PIR
- Priority - Priority of PIR when instantiated
- StartTime - Valid start time
- EndTime - Valid end time
- MasterId - Computer generated Id field
- SIRs - List of the Ids of associated SIRs

The SIRClass object has the following attributes:

- Id - Computer generated Id field
- Text - Text of SIR
- Nodes - List of the Ids of associated nodes

The NodeClass object has the following attributes:

- Id - Computer generated Id field
- Name - Generic unit type and size
- Description - Textual definition
- VisualSignature - List of imagery signatures
- COMINTSignature - List of communications signatures (radios)
- ELINTSignature - List of electronic signatures (radars)
- RADINTSignature - List of radioactive signatures

It is the asset attribute values and the nodal signature attribute values that are used in the analytical engine to determine whether an asset is capable of collecting against an intelligence requirement.

Another object that is necessary and is included in the PIR Hierarchy object chain is the Named Area of Interest (NAI) object. The NAI object represent an actual physical entity on the battlefield such as a bridge, hilltop, or road intersection. The NAI ties into the PIR hierarchical chain by associating with the SIRs. The NAIClass object has the following attributes:

- Id - Computer generated Id field
- Name - Text identification

- Description - Textual definition
- Coordinates - UTM grid coordinates
- SIRs - List of associated SIRClass Ids
- ExpTime - Expiration Time
- ExpEvent - Expiration Event

The object that ties all the Assets, PIRs, and NAIs together is the Tasking object.

If an asset is determined to be eligible for collection, the Tasking object is instantiated and used to maintain the specific information of which asset will collect against which PIR and at what location. The Tasking object has the following attributes:

- Id - Computer generated Id field
- Description - Textual definition
- EventTag -
- Criticality - Priority
- Status -
- Times - List of Start and End Times
- PIRTag - Id of associated PIRClass object
- NAITag - Id of associated NAI object
- AssetTag - Id of associated AssetClass object

F. DESIGN OF THE IEW SYNCHRONIZATION ANALYTIC ENGINE

Once the assets and PIRs (down to signatures) have been defined and selected for a specific COA, the Collection Manager must then evaluate the resources chosen. This entails matching prioritized requirements with suitable collection and exploitation assets using the following criteria:

1. Availability

A collection manager needs to know the assets at his/her own echelon, above and below. He/She also needs to know capabilities and how to assess them. Besides the maintenance and operator readiness issues, the intelligence staff officer has influence over the availability of organic assets to be available for collection and exploitation taskings.

2. Capability

An asset's capability criteria is fairly straightforward with electronic collection and exploitation systems. Capability includes such things as:

- Range (both actual distance and electromagnetic spectrum)
- Day and night effectiveness
- Technical characteristics
- Reporting timeliness
- Geolocation accuracy

3. Vulnerability

A collection manager must be able to evaluate the collection asset's vulnerability to threat forces. The threat's ability to locate, identify, and target the friendly asset must be considered.

4. Performance History

Within the intelligence community certain collection assets have a reputation of performing 'better' and are therefore more heavily relied upon to meet the intelligence requirements. Readiness rates, responsiveness, and accuracy over time may raise a collector's reliability quotient. [FM34-2]

The analytical engine of this thesis is comprised of eight algorithms (methods) developed to determine the capability, availability, and vulnerability of an asset. The algorithms are as follows:

- **checkUp** - The purpose of this method is to determine if the collection schedule of an asset overlaps the time that a PIR is valid. If there is no overlapping time, then the asset is not capable of collecting against a PIR and the asset's Up attribute is set to False and status attribute is set to 0, otherwise it is 'True' and 10.
- **checkCapable** - The purpose of this method is to determine whether the asset is able to acquire the target signature. If the asset type is either IMINT or HUMINT, the isCapable attribute is set to True and the status attribute is set to 10. If the asset type is COMINT, each value in the nodes' COMINTSignature list is compared to each value in the assets' CantCollect list. If each value in the COMINTSignature list is in the CantCollect list, then the isCapable attribute is set to 'False' and the status attribute is set to 0, otherwise it is 'True' and 10. The ELINT assets are compared in the same manner.

- **checkInRange** - The purpose of this method is to determine whether an asset is within range of a Named Area of Interest (NAI). The algorithm calculates the distance between a ground position or aerial center track line and center of mass of the NAI. If the calculated distance is greater than the doctrinal range of the asset, the `inRange` attribute is set to ‘False’ and status attribute is set to 0, otherwise it is ‘True’ and 10.
- **checkTimely** - The purpose of this method is to determine if an asset’s administrative processing (time it takes to provide intelligence) is greater than the Latest Time Intelligence Is Of Value (LTIOV). The algorithm calculates the Required Reporting Time as a sum of an asset’s TaskingLeadTime, ProcessingTime, and ReportingTime and compares the value to a PIR’s EndTime. If the total Required Reporting Time is greater (later) than the PIR EndTime, then the `isTimely` attribute is set to ‘False’ and the status attribute is set to 0, otherwise it is ‘True’ and 10.
- **checkMaxTaskings** - The purpose of this method is to determine if an asset has exceeded its maximum mission load amount. The size of an asset’s current tasking list is compared to its `MissionLoadMax` attribute value. If the number of taskings has exceeded the asset’s mission load capability, then the asset’s `MaxTaskings` attribute is set to ‘True’ and the status attribute is set to 0, otherwise it is ‘False’ and 10.
- **checkEnemyThreat** - The purpose of this method is to determine if an asset’s doctrinal threat vulnerabilities coincide with the actual AO threat vulnerabilities. Any overlapping vulnerabilities will degrade the asset’s capability status. For each identical vulnerability, the overall asset status is graded by a value of one and the `EnemyThreat` attribute will be set to ‘True’.
- **checkWeatherThreat** - The purpose of this method is to determine if an asset’s doctrinal weather vulnerabilities are the same or more severe than the actual AO weather forecast. The asset’s capability status will be degraded if its vulnerability values are equal to or greater than the current weather and the `WeatherThreat` attribute will be set to ‘True’.
- **checkLOS** - The purpose of this method is to check whether or not the asset has direct Line Of Sight (LOS) to the target. The map server that the IEW Synchronization matrix is built into will provide the LOS information upon query.

The result of each algorithm will set a boolean field in each instantiated asset object.

The cardinality of the ‘True’ values is assigned to the status attribute and evaluated as capable, marginal, not capable. This evaluation is done for each asset and PIR combination. Each of the criteria is addressed except for performance history. Currently, there is no mechanism that record asset performance in reference to responsiveness and accuracy. This knowledge normally resides in the resident subject-matter-expert.

G. DESIGN OF THE IEW SYNCHRONIZATION MATRIX GUI

The automated collection management tool, to be incorporated into the IEW Synchronization Matrix, provides the user with two main work spaces called the Asset Management and Requirements Management modules. These modules are designed to doctrinally resemble the CM Asset Management and Requirements Management subfunctions. The Mission Management module is built into the Requirements Management module.

1. Asset Management Module

Once the user invokes the IEW Synchronization Matrix program, the user will be provided the Asset Management screen main menu bar shown in Figure 10. The Asset Management window gives the user eight possible push-button choices. By pressing “Requirements” button the user can go into the Requirements Management Module. By pressing the “Plans” button, a submenu will be displayed allowing the user to either save a collection plan or to send the collection plan across a network to a subordinate or higher lever unit. By pressing the ‘Synchronized Time’ button, the user will invoke a system call to plot the current time on the timebar of the matrix canvas and to update the date and times of the system, H-hour, and battle day. By pressing the “Setup” button, the user will be able to set the H-hour time and date, set the sunrise, sunset, moonrise, moonset, Before Morning Nautical Twilight (BMNT), and End Evening Nautical Twilight. Additionally, the user will be able to add or delete a map overlay and initiate a dynamic map. The ‘Print Matrix’ button allows a user to print the AM window to a postscript printer. By pressing the ‘Other’ button, the user will be able to invoke the higher level synchronization matrix, OSM, the Deception Plan, or the Jamming Schedule (the latter two are not built as of yet and will be discussed in the last chapter). The ‘Exit ISM’ button exits the application and returns the user either to the base system. The ‘Help’ button invokes the Mosaic browser and allows the user to get a basic overview and a technical description of the application.

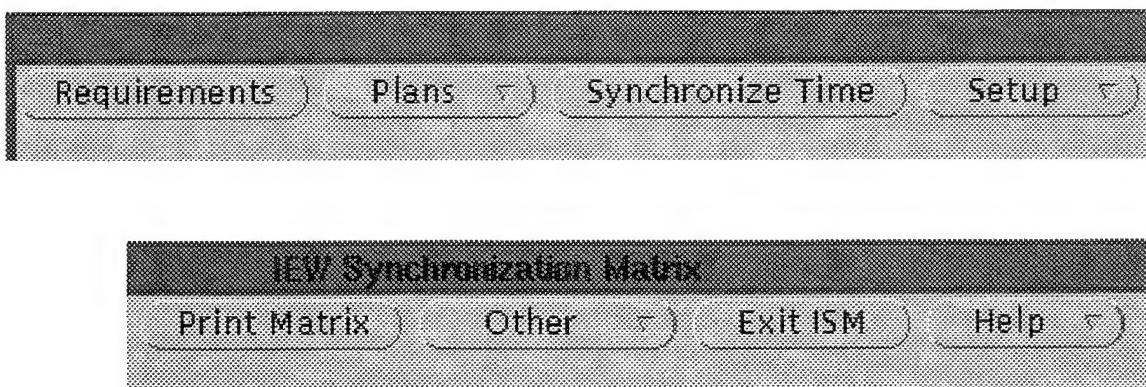


Figure 10: Asset Management Main Menu Bar

2. Requirements Management Module

By selecting the ‘Requirements’ button in the AM module, the user will have displayed a main menu bar as depicted in Figure 10. By pressing the “Plans” button, a submenu will be displayed allowing the use to either load a previously defined plan, delete an existing plan, or clear the canvas’ of a presently viewed plan.

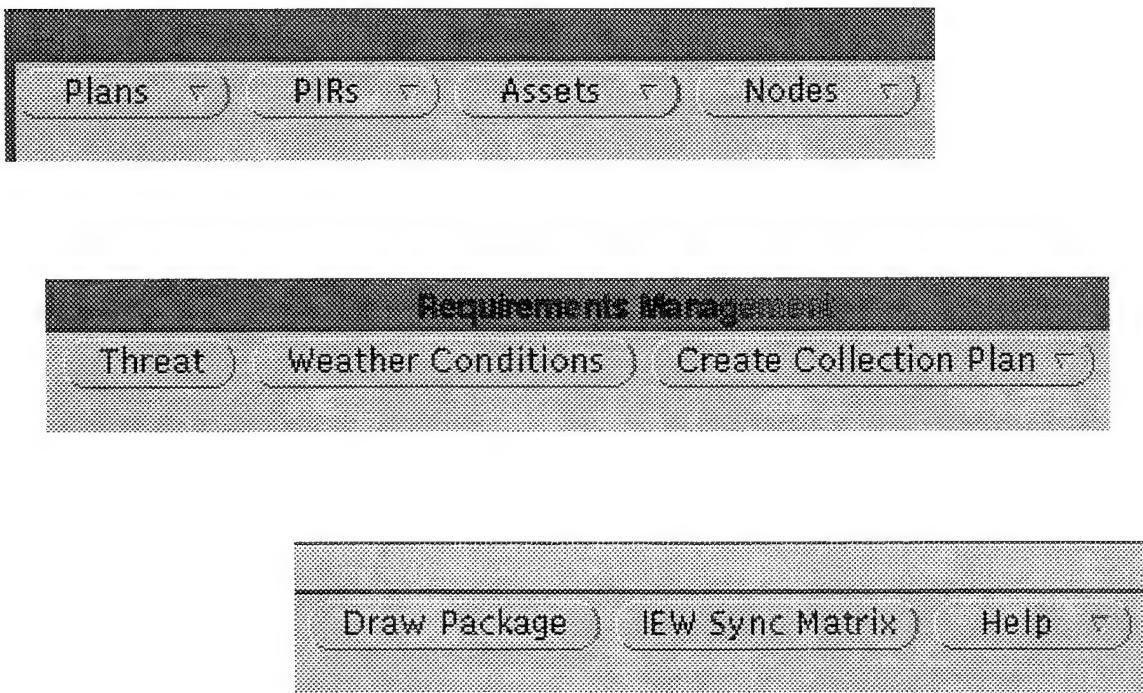


Figure 11: Requirements Management Main Module

The next three buttons in Figure 10: ‘PIRs’, ‘Assets’ and ‘Nodes’, are for database entry purposes. By pressing the ‘PIRs’ button, a submenu is displayed allowing the user to define the PIR-SIR-Node-Signature hierarchy or to delete the a specific PIR. By pressing the ‘Asset’ button, a submenu is displayed allowing the user to enter new asset data, edit exiting asset data, or delete assets from the database. The ‘Node’ button will give the user a choice of entering new node data to database or edit existing nodes in the database.

The next two buttons in Figure 10: ‘Threat’ and ‘Weather Conditions’, are used for setting the current AO vulnerabilities for a CM plan in development. The ‘Threat’ button gives the user a five option choice of threat vulnerabilities that commonly exist on the battlefield: Direct/Indirect Fires, Air Defense Artillery, Air to Air, Air to Ground, and Special Operations. The ‘Weather Conditions’ button allows the user to define the current AO environmental conditions in eight different categories:

- Cloud Cover (Setting and Description)
- Direction of Surface Winds
- Force of Surface Winds
- Visibility of the Surface
- Present Weather and Obstruction of Vision
- State of Roads
- State of Terrain
- State of Water Surface

A complete explanation of the weather categories and their possible values is given in Appendix C.

By pressing the next button, ‘Create Collection Plan’, the user is able to select assets and PIR’s for a collection plan created for a specific COA during the tactical decision making wargaming process or during real-time operations. The user will be able to select any number and type of available assets and uniquely define each by identification number and location. The user will also be able to select PIRs/SIRs combinations from the database, to prioritize PIRs, and to set the start and end times. The order of selection of the

assets and PIRs in creating the collection plan is not important. The analytical engine is invoked upon the selection of the second of the two. The user will then be provided in a matrix format (assets on y-axis and PIRs on x-axis) the results of the asset capability, availability, and vulnerability algorithms (See Appendix A).

The next button in Figure 10, ‘Draw Package’, is an on-line drawing table which gives the user the ability to conceptualize and to draw a plan using the mouse device on the window table displayed. The ‘IEW Sync Matrix’ button returns the user to the Asset Management module and automatically loads the current collection plan if one exists. The ‘Help’ button invokes the Mosaic browser and allows the user to get a basic overview and a technical description of the application.

H. SUMMARY

The design of the system structure and interface of the Collection Management Tool is modular and efficient. Most importantly, the intelligence analyst is now able to better support a commander both in wargaming and real-time operations. The system structure object design of the asset and PIR hierarchy was very easily solved using the OO paradigm. The analytical engine is comprised of eight algorithms that test for sensor capability, availability, and vulnerability, and provides all the necessary tests to determine whether an asset can collect against a PIR, but in a fraction of time. The CM GUI is an easy use interface that closely resembles the doctrinal format in U.S. Army field manuals. Additionally, the code can be reused by other Battlefield Operating Systems in the design of their synchronized operations.

VI. CONCLUSIONS

A. INTRODUCTION

With the onset of advanced technology weapons systems on today's battlefield, it is imperative that an automated tool be developed to manage these systems. This thesis represents the initiation of a major prototyping effort to automate the way a commander and intelligence staff manage collection assets in a wargaming environment and on the battlefield. The primary focus of this work is to design and implement the Requirements Management and Mission Management Modules as part of the Collection Management Tool in the IEW Synchronization Matrix.

B. EVALUATION

1. Collection Management Tool GUI

The GUI developed in this thesis is a logically structured and doctrinally correct tool that allows an intelligence analyst to flow through the Collection Management cycle and its intertwined sub-functions. The GUI is easy to use because the automated visual design mimics the manual process currently in use and therefore the learning curve for users is minimal. A user can be completely familiar and having a working collection and asset plan in minutes. This is a marked improvement over the old manual way which is on the order of hours in processing.

2. AIE Language Code

The AIE language code is a high level OO programming environment which is very easy to learn and to work. The language only consists of objects, their attributes and methods and forces the designer/programmer to completely design and solve an issue in an OO manner. The reference and example manuals available are very clear and give excellent explanations as to how to begin development. A fairly strong background in C++

programming and an understanding of the OO paradigm is required and the developer is able to begin working in the AIE environment.

C. FUTURE WORK AREAS

1. Collection Management Tool

The automated CM tool is presently in a working prototype status. There are however, a few enhancements that will make the tool more doctrinally realistic and improve the asset-PIR evaluations.

a. *By-Echelon Tailoring*

Currently, the automated CM tool allows a user to task all assets available to him from battalion up through theater level. A mechanism needs to be incorporated to check the asset ownership of the user at each echelon and only allow the user to task those assets that are directly under his control. All other tasking requirements would then be interpreted as a request for tasking. By tailoring the product to each echelon a user will not be given a false sense of intelligence collection support.

b. *Asset Scheduler*

Currently, the collection asset schedules are loaded from the host database. The GUI needs to be enhanced to allow the Operational or Administrative asset owner to enter an asset's collection and maintenance schedule. This will make the tool more dynamic, flexible, and give the user more of a feeling of control over the system.

c. *Performance History Evaluation Algorithms*

A mechanism needs to be built into the CM tool that maintains a historical database on an asset's collection readiness status, responsiveness to taskings, and accuracy of collection. This data can then be used by a performance history evaluating algorithm that can give a probabilistic determination as to an asset's capability.

2. IEW Synchronization Matrix

The IEW Synchronization matrix is comprised of three functions. One function, the Collection Management process, was dealt with in depth in this thesis. The other two functions are Jamming and Deception. In order for a Commander to get a complete intelligence picture of the battlefield, not only does he need to see that his AO has complete collection coverage, but also that there are active offensive plans to deceive and obstruct the enemy from obtaining any information about friendly operations.

a. Jamming Schedule

A jamming schedule is an active offensive operation that is used to prevent the enemy from electronically collecting against friendly forces during tactical operations. The automated jamming schedule should depict in a matrix format the assets and times of jamming. The jamming schedule will be coordinated with and support the deception plan and friendly maneuvers. Additionally, the automated jamming schedule is critical and will be closely coordinated with the collection schedule so that valuable targeted enemy units can be exploited and obstructed at the right times. By incorporating the jamming schedule into the IEW Synchronization Matrix a Commander will be given a more complete picture of the battlefield.

b. Deception Plan

A deception plan is a passive offensive operation that is used to feign the enemy into believing that friendly forces are conducting a certain tactical maneuver when in fact they are not. Developing a deception plan is a jointly coordinated action between the Operations and Intelligence staffs. The deception plan must coincide with an actual operational maneuver and be successful in diverting attention to it. Automating this process and incorporating it into the IEW Synchronization Matrix will give a Commander a more complete picture of the battlefield.

3. Artificial Intelligence Applications

The addition of multiple AI technologies to the IEW Synchronization Matrix will enhance the capability of commanders and their intelligence staffs in performing their intelligence tasks. There are several decision support system implementations that can readily be built into the IEW Synchronization Matrix software.

a. Implementation of Intelligent Event and Decision Support Templates

The use of cased based reasoning techniques will allow the development and collection of object oriented models. These models can be incorporated into event and decision support templates, to be used in both the battle planning and battle execution phases.

b. Named Area of Interest Monitoring

NAI event monitoring can provide automation support to the mapping of incoming intelligence data with outstanding information requests. Using a nested arrangement of rule based and case based reasoning, the event monitors will supply a hierarchical infrastructure from which the states of the NAI and the events, targets, and collectors within the NAI can be dynamically and automatically ascertained.

c. Self-Monitoring Decision Points

Creating software daemons to describe and then look to confirm specific events will greatly enhance the IEW Synchronization Matrix operator's ability to react to unfolding battlefield events. The automated decision points, while gathering their own evidence to support or deny their own existence, will make suggestions to support the decision making process. [BCBL95]

D. SUMMATION

Army Intelligence, by definition and design, focuses on support to the tactical warfighter in a dynamically-based frontier. By automating a major function of the U.S. Army intelligence process, the intelligence soldiers are now able to provide a more timely

and accurate picture of the battlefield. This will allow full integration of Military Intelligence as an effective Battlefield Operating System in support of the tactical warfighter. This project is in line with the U. S. Army Force XXI design efforts. It has taken current U. S. Army doctrine and automated it thereby ensuring soldiers are able to operate in a more timely, efficient, and manner. This will give a commander more time to make the necessary critical designs on today's dynamic battlefield. This thesis is totally relevant to the U. S. Army and is currently fielded to units in Korea; Fort Drum, New York; Fort Hood, Texas; and the Army Research Lab (ARL), Adelphi, Maryland. Additionally, the National Security Agency, Fort Meade, Maryland has a working copy for evaluation for use at the strategic level. The software was demonstrated and received accolades from the U.S. Army Chief of Staff and Chief of Technical Operations at the Spring '95 AUSA conference, Santa Clara, California. In August, 1995, the software was demonstrated and installed for testing at the DoD agency Advanced Research Project Agency (ARPA), the National Reconnaissance Office (NRO), PM All Source Analysis System (ASAS), PM Joint Collection Management Tool (JCMT). The Operations Support Office, NRO has expressed an interest in fielding this CM tool with an off-the-shelf software product called Satellite Tool Kit (STK) and the PM, JCMT, has also expressed an interest in incorporating the CM tool in its project line. The STRICOM, University of Central Florida, and West Point have all shown interest in the product. The visibility and interest shown so far can only be indicators that the development of an automated collection management tool is necessary and will ultimately be funded as a U.S. Army requirement.

APPENDIX A. SYNCHRONIZATION MATRIX USER'S GUIDE

This appendix contains a detailed and complete users manual for all the collection management functions of the IEW Synchronization Matrix.

A. STARTING AND EXITING THE APPLICATION

The IEW Synchronization Matrix is capable of running as an integrated part of ASAS-Warrior and as a stand-alone application. This users manual will provide start-up instructions for the stand-alone version. To initiate the IEW Synchronization Matrix, the user must be in the proper directory. The application is started at the command prompt by entering:

> ismX

To quit the Collection Management Tool, select the “Exit ISM” button on the main menu bar of the Asset Management module canvas.

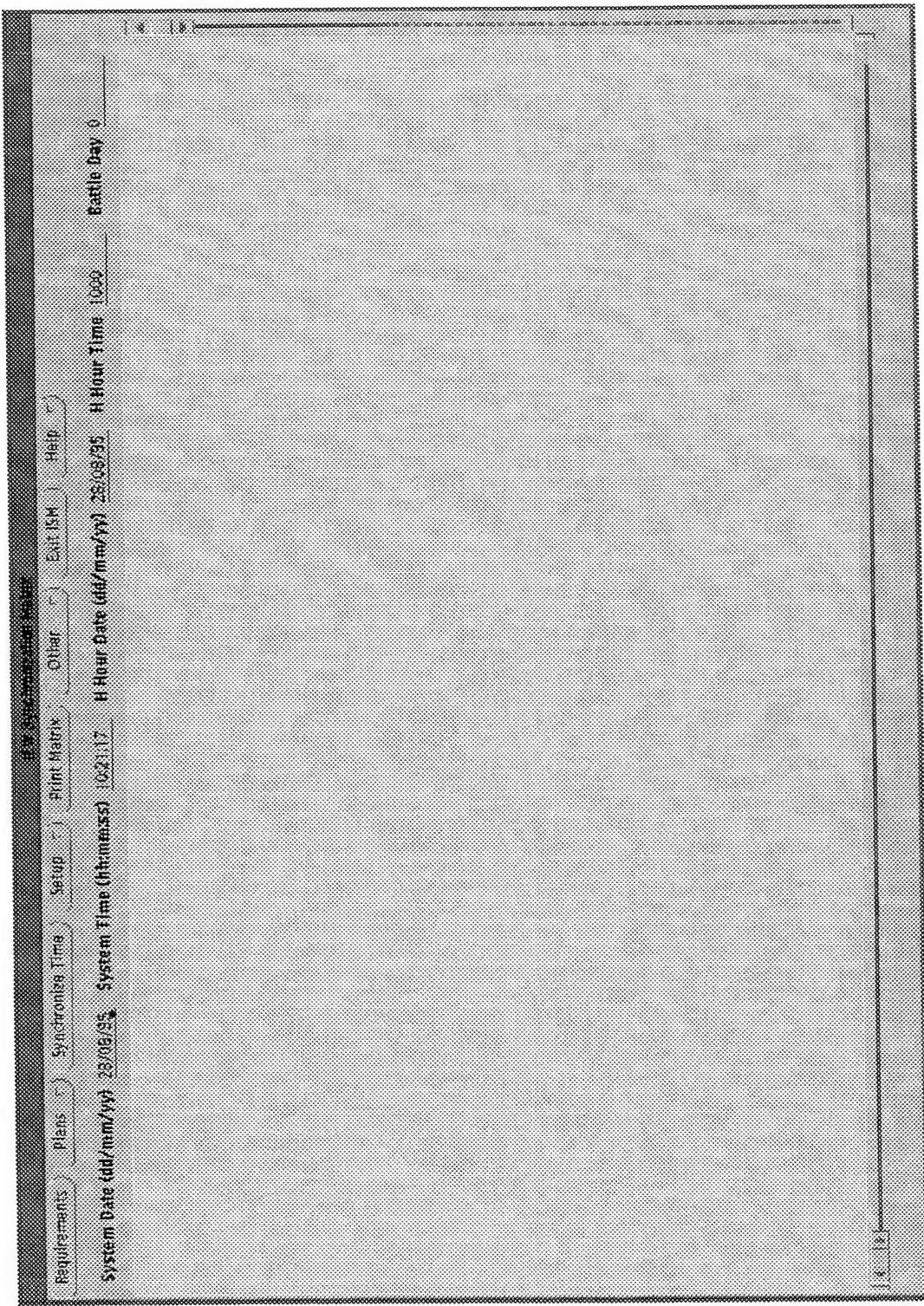


Figure A-1: Asset Management Module Window

B. WORKING IN THE ASSET MANAGEMENT MODULE

The initializing window is the Asset Management module as shown in Figure A-1. The AM module has a single main function: to management collection assets' schedules and taskings. The AM window has a main menu button bar and an absolute and relative time/day information bar across the top and an empty canvas screen below. When a collection plan is loaded, the canvas displays a matrix of collection assets' schedules and taskings across a relative time bar.

1. Loading the Requirements Management Module

To initialize the RM module, select the “Requirements” button. This is the first step in creating a collection plan. From this point, go to Section C. of this appendix, Working in the Requirements Management Module.

2. Plans

Once a collection plan has been created in the RM module, the user now can return to the AM module and either save a plan or to send the plan to another unit. Select the “Plan” button on the main menu bar with the right mouse button. A submenu will be displayed with the choices to “Save Plan” or “Send Plan”. The submenu choice is made with the left mouse button. To save a plan, select the “Save Plan” option and the window in Figure A-2 is displayed. Input a plan name with no spaces at the prompt and select the “Save” button to save the plan. If the “Save Plan” button is erroneously selected, press the “Cancel” button to close the window. No changes will be made to the system.

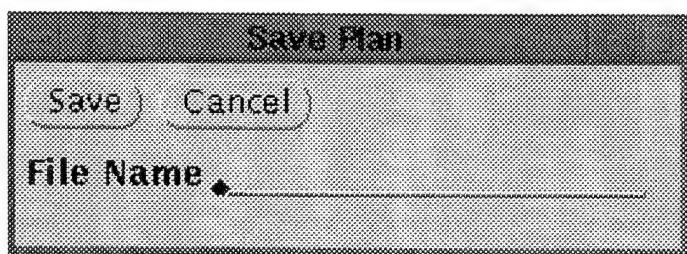


Figure A-2: AM Module - Save Plan Window

To send a plan, select the “Send Plan” option and the window in Figure A-3 is displayed. Input a plan name and host name each with no spaces at the prompt and select the “Send” button to forward the plan. If the “Send Plan” button is erroneously selected, press the “Cancel” button to close the window. No changes will be made to the system.

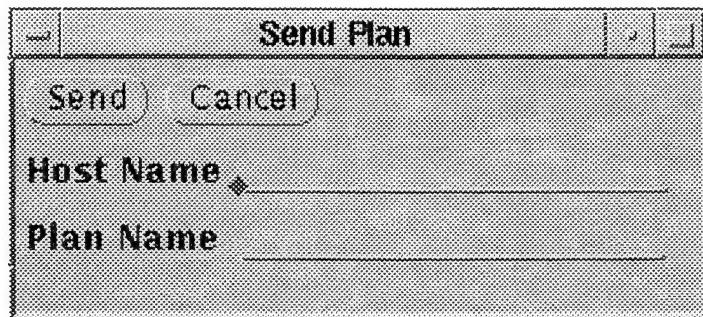


Figure A-3: AM Module - Send Plan Windows

3. Synchronize Time

To synchronize the absolute and relative times, select the “Synchronize Time” button. This will invoke a system call to plot the current clock time on the timebar of the matrix canvas and to update the date and relative times of the system, H-hour, and battle day.

4. SetUp

To manually set the time and environmental factors select the “Setup” button on the main menu. A Matrix Control Panel window in Figure A-4 will be displayed. Use slider bars to manually set the operational start time and to set the H-hour time and date. The sunrise, sunset, moonrise, moonset, Before Morning Nautical Twilight (BMNT), and End Evening Nautical Twilight (EENT) can also be set via a slider bar relative to H-hour.

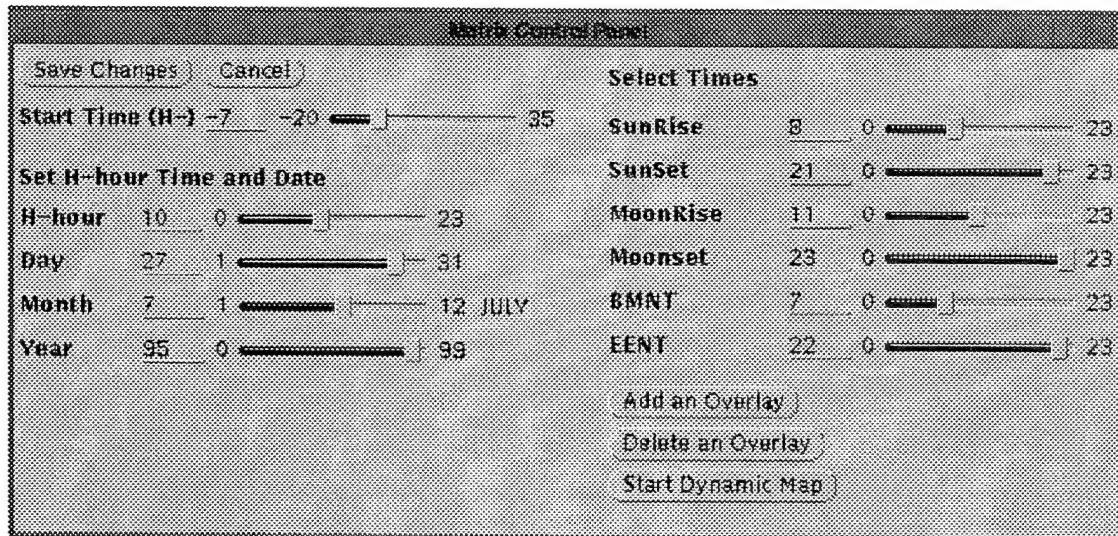


Figure A-4: AM Module - Matrix Control Panel

To add an overlay, press the “Add an Overlay” button. A window will be displayed with map overlay choices to select from. Select a choice and the map overlay will be displayed. To delete an overlay, press the “Delete an Overlay” button. A window will be displayed with map overlay choices to select from. Select a choice and the map overlay will be removed from the database. To start a dynamic map from a map server, press the “Start Dynamic Map” button. Select the “Save Changes” button to exit the window and save the time and environmental factors. If the “Setup” button is erroneously selected, press the “Cancel” button to close the window. No changes will be made to the system.

5. Print Matrix

To print the matrix canvas with asset scheduling and tasking data, select the “Print Matrix” button. This will invoke a system call to print the data in raster format.

6. Other

To switch to the Operational Synchronization Matrix (OSM) or to one of the other IEW Synchronization functions, Jamming Schedule and Deception Plan, select the “Other” button with the right mouse button. A submenu will be displayed with the choices

to “OSM”, “Deception”, “Jamming”. The submenu choice is made with the left mouse button.

7. Help

To get help or information about the matrices, select the “Help” button with the right mouse button. A submenu with the different synchronization matrices will be displayed as in Figure A-5. Select the synchronization matrix of choice with the left mouse button. The WWW browser, Mosaic, will be invoked and display the help information.

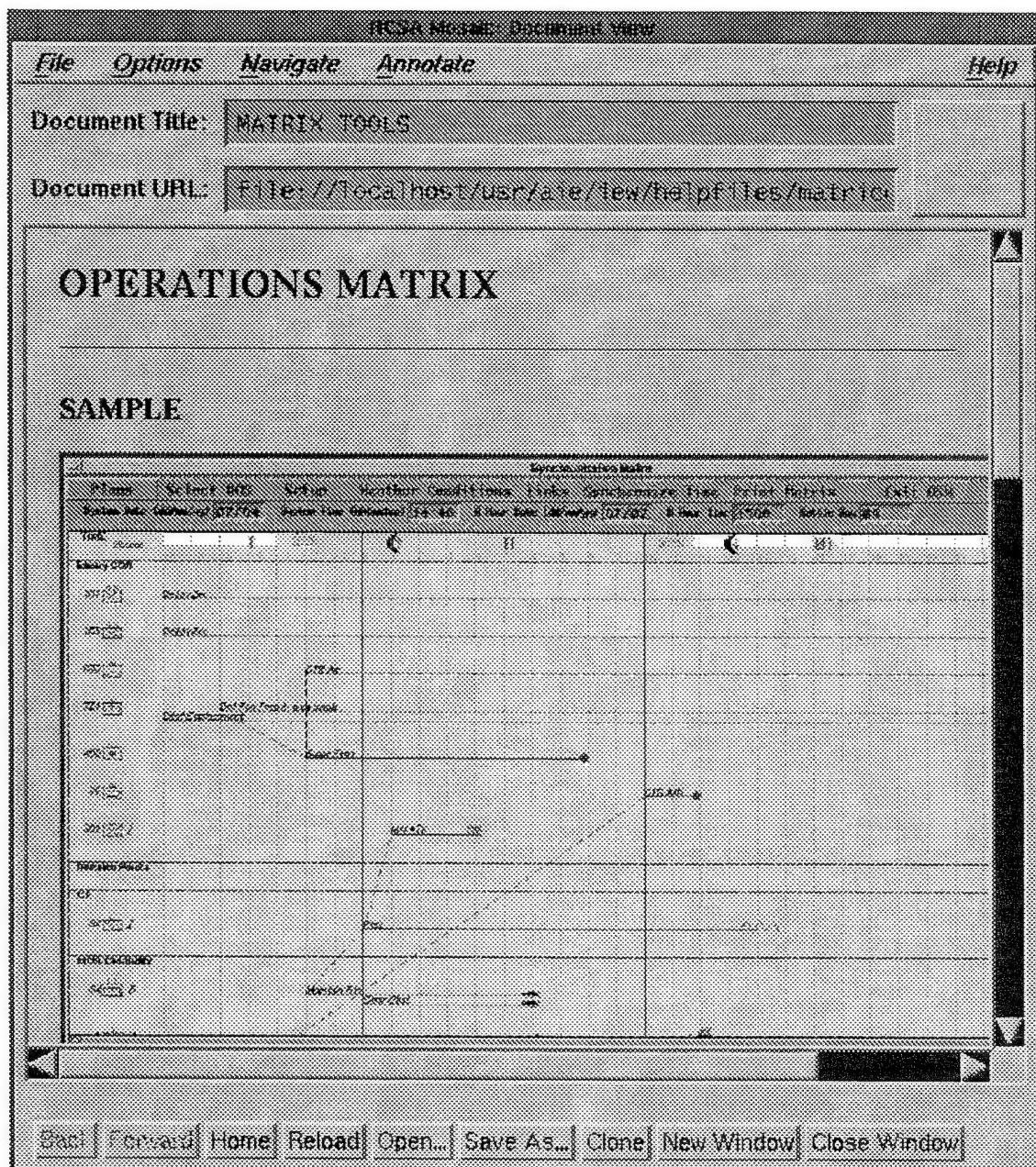


Figure A-5: Mosaic Help Page

C. WORKING IN THE REQUIREMENTS MANAGEMENT MODULE

The RM module has multiple functions: to enter assets, PIRs, Indicators (SIRs), Nodes, and Signatures into the database, to create collection plans to support a COA, and to load a previously defined collection plan for operational purposes. The initial RM module window is shown in Figure A-6. The RM window has a main menu button bar, an upper canvas, and a lower canvas. The upper canvas is used to display the PIRs, Indicators (SIRs), Nodes, and Signature text. The lower canvas is used to display the assets and PIRs selected for a particular collection plan. To quit the RM module, select the “IEW Sync Matrix” button. The user will be returned to the AM module. If a collection plan is created it will automatically be loaded into the AM module canvas. An example of a plan is shown in Figure A-35.

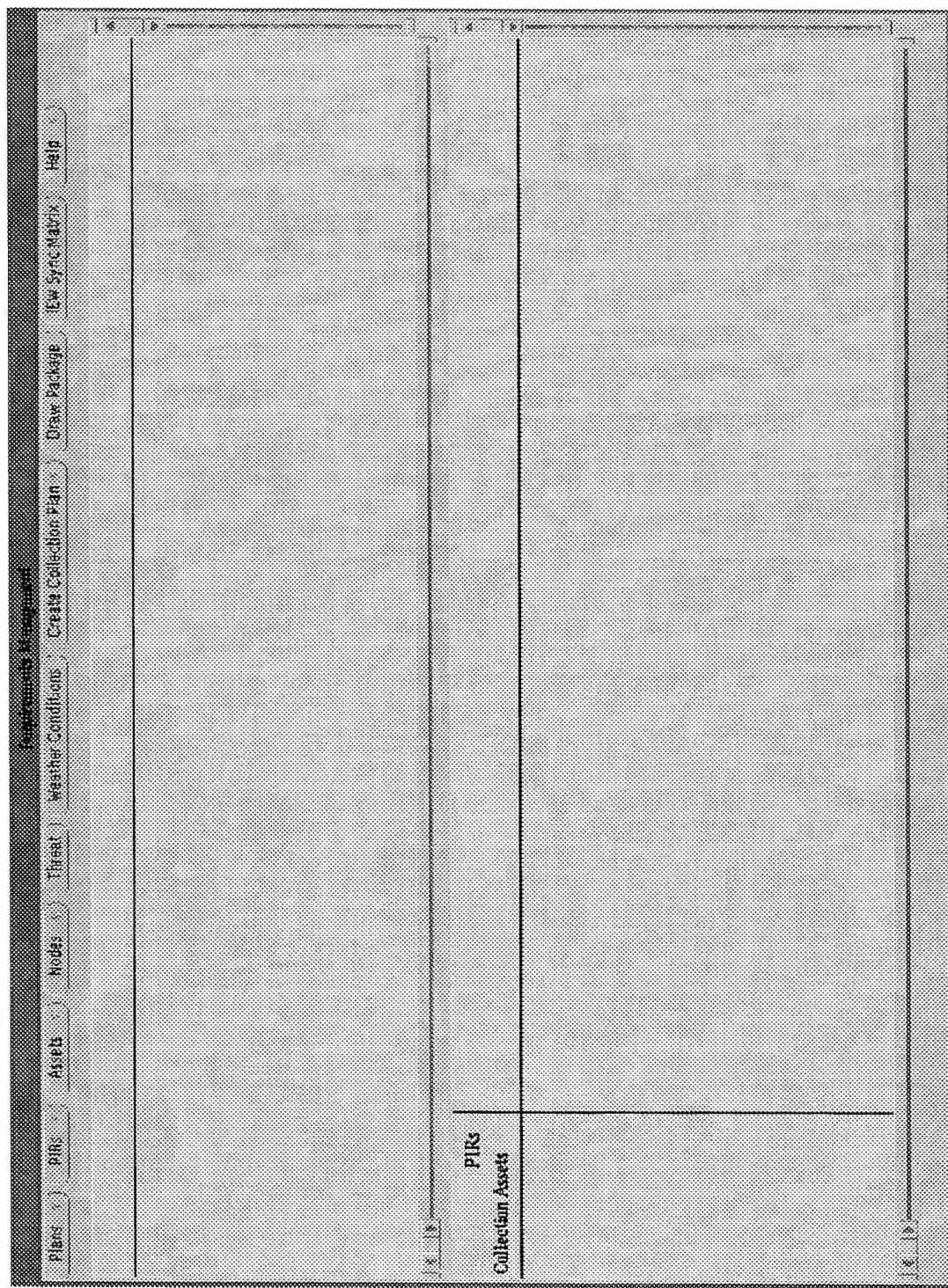


Figure A-6: Requirements Management Module Window

1. Plans

Once a collection plan has been created and saved, the plan can be reloaded, cleared, or deleted for operational purposes or for modifications. Select the “Plan” button on the main menu bar with the right mouse button. A submenu will be displayed with the choices to “Load Plan”, “Clear Plan”, or “Delete Plan”.

To load a collection plan, select the “Load Plan” submenu choice with the left mouse button. A ‘Load Plan’ window is displayed as in Figure A-7. Select a plan by pressing the button to the left of the plan’s name. If the submenu choice ‘Load Plan’ is erroneously selected, press the “Cancel” button to close the window. No changes will be made to the system.

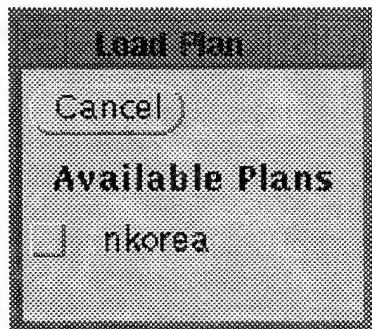


Figure A-7: RM Module - Load Plan Window

To clear a collection plan from the screen, select the “Clear Plan” submenu choice with the left mouse button. The upper and lower canvas’ will be cleared and a new plan can be loaded or created.

To delete a collection plan, select the “Delete Plan” button with the left mouse button. A ‘Delete Plan’ window is displayed as in Figure A-8. Select a plan by pressing the button to the left of the plan’s name. If the submenu choice ‘Delete Plan’ is erroneously selected, press the “Cancel” button to close the window. No changes will be made to the system.



Figure A-8: RM Module - Delete Plan Window

2. Entering Data into the Database

Before any collection plans can be created there must be asset and PIR data loaded into the database. This section instructs the user on how to enter, edit, and delete data from the database.

a. Assets

Select the “Assets” button from the RM module using the right left mouse button. A submenu with three choices will be displayed: “Enter New”, “Edit”, and “Delete”.

(1) Adding Assets - To add an asset, select the “Enter New” choice from the submenu of “Assets” using the left mouse button. An “Add Asset” window is displayed as shown in Figure A-9. is displayed. Input the attribute values for the asset at each prompt line. The application will not allow duplicate assets to be entered. To define the asset’s threat vulnerability, press the “Set Threats to Asset” option. A ‘Threat Vulnerability’ window is displayed as in Figure A-20. Select the asset’s threat vulnerabilities and press “Done” to save and exit threat window. To define the asset’s weather conditions vulnerability, press the “Set Weather Effects” option. A ‘Weather Forecast’ window is displayed as in Figure A-21. Use the slider bars to set the asset’s weather vulnerability and press “Done” to save and exit weather window. Press the “Add Asset” button after each

asset data is entered. This will save the data to the database and clear the attribute values. This allows for multiple asset data entry. When all asset data has been entered, select the “Done” button to exit the ‘Asset Information’ window.

The screenshot shows a software interface titled "RM Module - Add Asset". At the top, there is a toolbar with icons for "New", "Open", "Save", "Print", "Exit", and "Help". Below the toolbar, a menu bar includes "File", "Edit", "View", "Tools", and "Help". A sub-menu under "File" lists "New", "Open", "Save", "Print", "Exit", and "Help". The main window contains several sections:

- Asset Vulnerabilities**:
 - Set Weather Effects
 - Set Threats to Asset
- Instance Name:** Nil
- Classification:** Nil
- Caveats:** Nil
- Compartments:** Nil
- Owner:** Nil
- Pretty Name:** Nil
- Asset Type:** Nil
- Frequency Minimum:** Nil
- Frequency Maximum:** Nil
- Modulation:** Nil
- Standoff:** Nil
- Tasking Lead Time:** Nil
- Processing Time:** Nil
- Reporting Time:** Nil
- Power:** Nil
- MTBF:** Nil
- MTTR:** Nil
- Mission Load Maximum:** Nil
- Endurance:** Nil
- Report Type:** Nil
- Can't Collect:** Nil
- Imint Resolution:** Nil
- Catalog:** Nil
- Does Range Effect:** Nil
- Does LOS Effect:** Nil
- Range:** Nil
- Platform Data:** Nil
- Remarks:** Nil

Figure A-9: RM Module - Add Asset Window

(2) Editing Assets - To add an asset, select the “Edit” button from the submenu of ‘Assets’ using the left mouse button. An ‘Edit Asset’ window is displayed as in Figure A-10a. Select an asset from the choice column and an ‘Asset Information’ window is displayed as in Figure A-10b. Edit the appropriate asset attribute fields and select the “Save Changes” button to save the changes to the data base. To exit the window, press the “Quit” button. The ‘Edit Asset’ window is still active which allows multiple asset editing. Select another asset form the choice column or press the “Done” button to exit the window.

Edit Assets	
Date: <input type="text" value="2023-09-15" style="width: 100px; height: 20px; border: 1px solid black; border-radius: 5px; padding: 2px; margin-right: 10px; font-size: 10px; font-family: monospace; border-bottom: none;">	
Select an Asset to Edit	
<input checked="" type="checkbox"/> LRS TEAM <input type="checkbox"/> CT TEAM <input type="checkbox"/> AIG-133 <input type="checkbox"/> USG-SA <input type="checkbox"/> AIG-151V2 <input type="checkbox"/> TSD-1142H1 <input type="checkbox"/> TBC-90 <input type="checkbox"/> TBC-329V1 <input type="checkbox"/> ISTARS <input type="checkbox"/> UAV-MR <input type="checkbox"/> CBCS-SINT <input type="checkbox"/> TPO-J7 <input type="checkbox"/> AEROSCATTS <input type="checkbox"/> FAAR <input type="checkbox"/> TPS-58 <input type="checkbox"/> CBCS-COMINT <input type="checkbox"/> PPS-15AV1 <input type="checkbox"/> PPS-58 <input type="checkbox"/> CSD-187 <input type="checkbox"/> INTC-TEAM <input type="checkbox"/> ARTY-RC <input type="checkbox"/> RECON-PATROL <input type="checkbox"/> TPO-3S	
Asset Information <input style="margin-right: 10px; border: 1px solid black; border-radius: 5px; padding: 2px 10px; font-size: 10px; font-family: monospace; border-bottom: none; background-color: #f0f0f0; color: black; text-decoration: none; font-weight: bold; text-align: center; width: 100px; height: 20px; margin-bottom: 5px;" type="button" value="Save Changes"/> <input style="border: 1px solid black; border-radius: 5px; padding: 2px 10px; font-size: 10px; font-family: monospace; border-bottom: none; background-color: #f0f0f0; color: black; text-decoration: none; font-weight: bold; text-align: center; width: 100px; height: 20px;" type="button" value="Quit"/> Asset Vulnerabilities <input style="border: 1px solid black; border-radius: 5px; padding: 2px 10px; font-size: 10px; font-family: monospace; border-bottom: none; background-color: #f0f0f0; color: black; text-decoration: none; font-weight: bold; text-align: center; width: 100px; height: 20px; margin-bottom: 5px;" type="button" value="Set Weather Effects"/> <input style="border: 1px solid black; border-radius: 5px; padding: 2px 10px; font-size: 10px; font-family: monospace; border-bottom: none; background-color: #f0f0f0; color: black; text-decoration: none; font-weight: bold; text-align: center; width: 100px; height: 20px;" type="button" value="Set Threats to Asset"/>	
Instance Name: LRS TEAM Classification: U Caveats: Nil Compartments: Nil Owner: DIV CORPS Pretty Name: LRSQ Asset Type: HUMINT Frequency Minimum: Nil Frequency Maximum: Nil Modulation: Nil Standoff: 0 Tasking Lead Time: 12.000000 Processing Time: 0.100000 Reporting Time: 3.000000 Power: Nil MTBF: 0 MTTR: 0 Mission Load Maximum: Nil Endurance: 8 Report Type: SPOT REPORT Can't Collect: Nil Imint Resolution: Catalog: Does Range Effect: Nil Does LOS Effect: GROUND Range: T Platform Data: T Remarks: 10	

(a)

(b)

Figure A-10: RM Module - Edit Asset Choice

(3) Deleting Assets - To delete an asset, select the “Delete” choice from the submenu of “Assets” using the left mouse button. A ‘Delete Assets’ window is displayed as in Figure A-11. Select an asset, one at a time, from the choice column and press the “Delete” button. The window will be redisplayed with the revised master asset list. Select the “Quit” button to close the window.

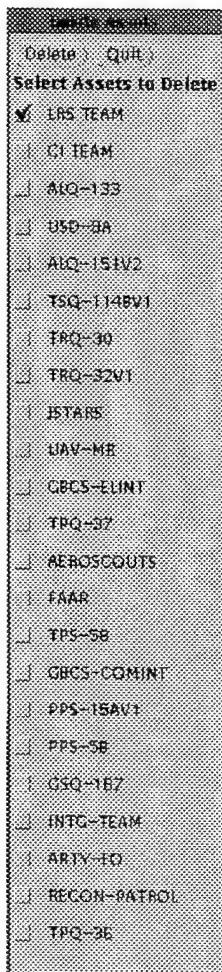


Figure A-11: RM Module - Delete Asset Selection

b. PIRs and SIRs (Indicators)

To enter PIR hierarchical information into the database, select the “PIRs” button from the RM module using the right left mouse button. A submenu with two choices

will be displayed: “Define PIRs/SIRs/Nodes” and “Delete”. The first submenu choice is for creating, editing and defining PIRs and SIRs; and deleting SIRs. The second submenu choice is for deleting PIRs. This will remove the entire hierarchical PIR definition chain.

To create or edit a PIR and define its hierarchy, select the submenu choice ‘Define PIRs/SIRs/Nodes’ with the left mouse button. An ‘Edit an Existing PIR’ window will be displayed as in Figure A-12.

<p>FIRs</p> <p>Save Changes Cancel</p> <p>Create New PIR Change PIR Text</p>	<p>When will Second Echelon regiment (25th Army) of the 11th Corps be committed?</p> <p>When will the enemy conduct the Main Attack along AOA B?</p> <p>Is there any evidence of NBC operations via PIR Blue?</p> <p>What are the locations of Army logistical units and forward distribution points?</p> <p>Will the enemy attempt to cross the river vicinity of AOA Purple?</p> <p>Other end where will the SGP fit across River Purple?</p>	<p>Create New SIR</p> <p>Click to view nodes click to Delete SIR</p>
<p>SIRs</p>	<p>Activation and forward lateral movement of regtment forward CP's</p> <p><input checked="" type="checkbox"/> Forward/lateral movement of 11th Corps forward CP</p> <p><input checked="" type="checkbox"/> Crossing of armor of major transportation points</p> <p><input checked="" type="checkbox"/> Forward/lateral movement of Div Regt PCL Ammo points</p> <p><input type="checkbox"/> Sustained artillery attack w/o immediate follow-up ground attack</p> <p><input checked="" type="checkbox"/> Activation of mainauer regt airborne CP's</p> <p><input type="checkbox"/> Reinforcement arriving at forward assembly areas</p>	

Figure A-12: RM Module - PIR Hierarchy Definition

To add a new PIR to the database, select the “Create New PIR” button. A ‘Create New PIR’ window is displayed as in Figure A-13. Type the PIR text at the prompt and press the “Save” button. The window will close and the newly created PIR will be displayed in the ‘Edit an Existing PIR’ window. Select the “Cancel” button to exit the window with no changes to the system.

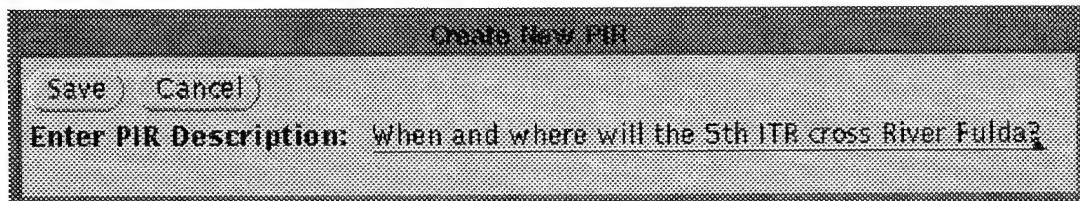


Figure A-13: RM Module - Add New PIR Window

Once a PIR is created, associated SIRs (Indicators) and Nodes must be defined for the PIR. To select associated SIRs, press the box from the choice column to the left of the SIR text. This will tag the SIR to be part of the PIR hierarchy chain. To get nodal information on a particular SIR, press the box in the ‘Click to View Nodes’ column to the right of the corresponding SIR. An ‘Edit SIR’ window will be displayed as in Figure A-14. The nodes associated with the SIR are checked to the left of the node.

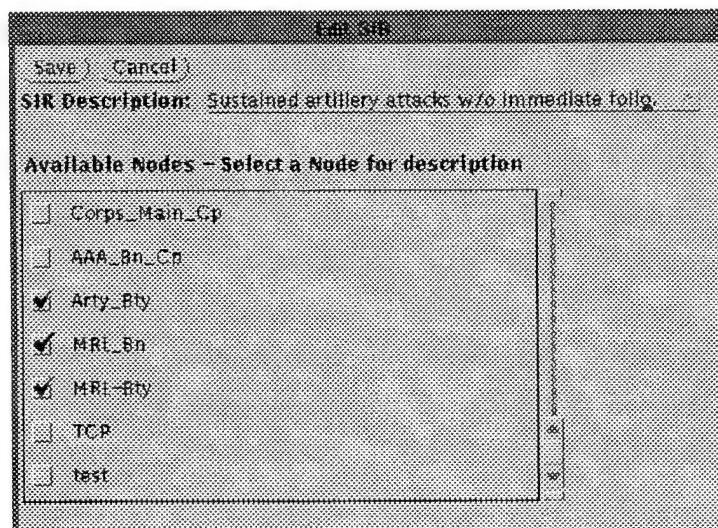


Figure A-14: RM Module - View SIR-Node Window

To add or delete a node from the PIR hierarchical chain and to obtain detailed information about a node, press the box from the choice column to the left of a corresponding node. The ‘Node Information’ window is displayed as in Figure A-15. To add the node to the SIR association list, press the “Select” button. To remove the node from the SIR association list, press the “Unselect” button. Press the “Cancel” to close the window and make no changes.

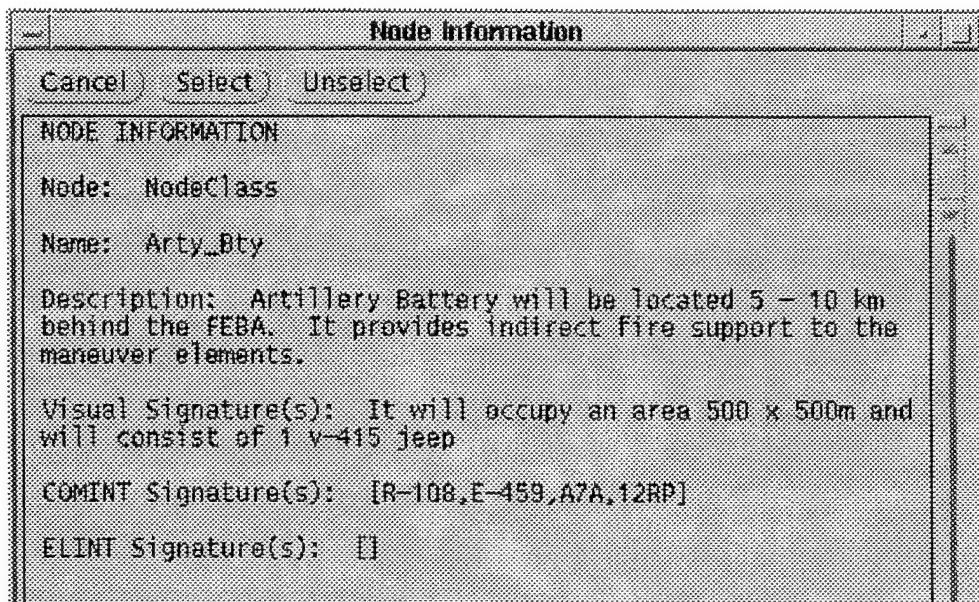


Figure A-15: RM Module - Node Information Window

To edit the text of a PIR, select a PIR by pressing the box to the left of the PIR text and then select the “Change PIR Text” button. A ‘Change PIR Text’ window will be displayed as in Figure A-16 with the PIR text filled in at the prompt. Change the text as necessary and press the “Save” button. The window will close and the edited PIR will be displayed in the ‘Edit an Existing PIR’ window. To edit the PIR hierarchy, flow through the same procedure as creating a PIR hierarchy by selecting or unselecting the boxes to the left of the corresponding SIR and Node.

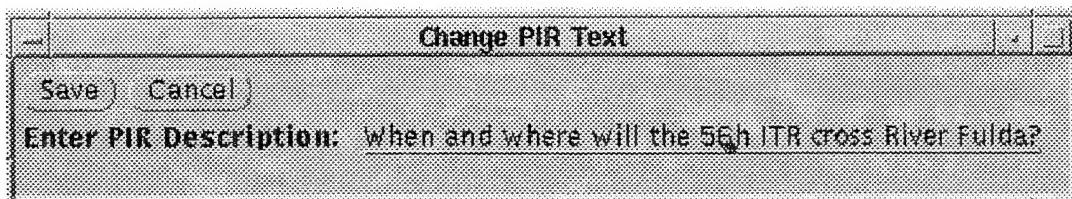


Figure A-16: RM Module - Change PIR Text

To add an SIR (Indicator) to the database select the “Create New SIR” button from the ‘Edit and Existing PIR’ window. A ‘Create New SIR’ window will be displayed as in Figure A-17. Enter the new SIR text at the prompt and select the associated nodes. Nodal information is as discussed previously in creating and defining a PIR. Press the “Save” button to close the window, save the changes, and add the newly created SIR to the database. The new SIR is displayed in the SIR table of the ‘Edit and Existing PIR’ window.

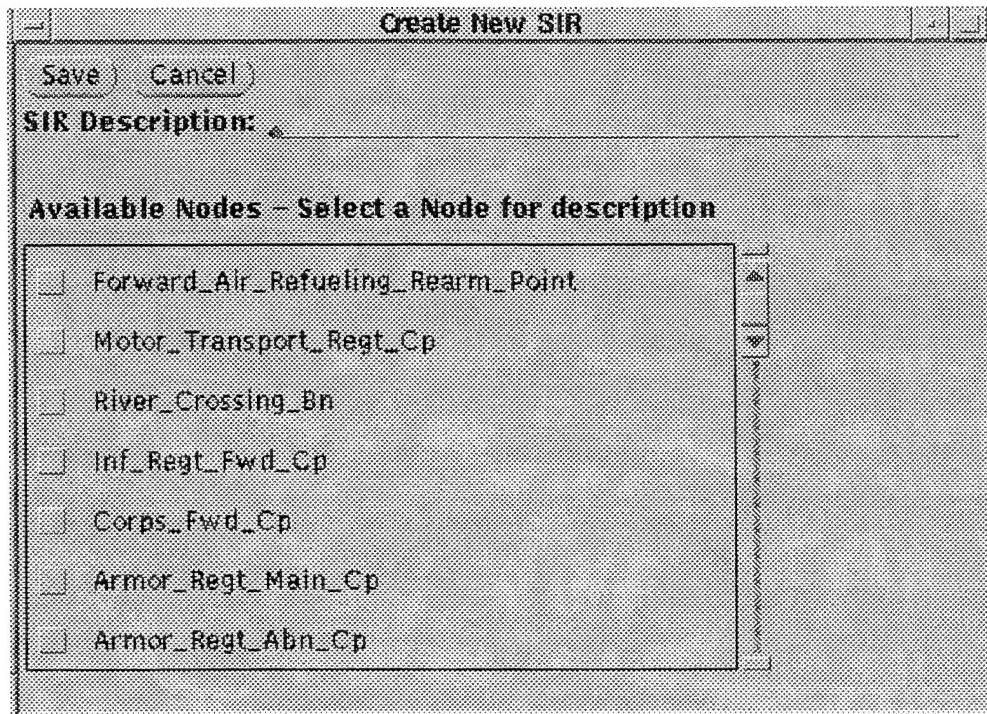


Figure A-17: RM Module - Create New SIR Window

To delete an SIR from the master database, press the box to the left of the SIR text to denote selection and then press the box in the ‘Click to Delete SIR’ column on the far right of the corresponding SIR text. A verification of intent will be displayed as in Figure A-18. To delete the SIR, press the “OK” button and the SIR will be removed from the database. To cancel the procedure, press the “Cancel” button. No changes will be made to the system.

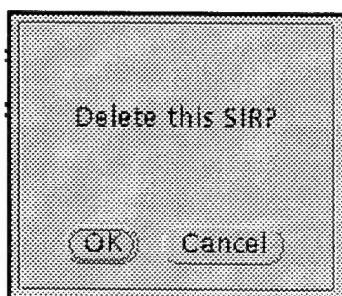


Figure A-18: RM Module - Verification Notice

When all PIR and SIR information have been entered into the database, select the “Select Changes” button to close the window and save the changes to the database files. If the ‘Edit an Existing PIR’ window is erroneously invoked, press the “Cancel” button to close the window with no changes to the system.

To delete a PIR from the master database, select the submenu choice ‘Delete’ with the left mouse button. A ‘Delete PIRs’ window will be displayed as in Figure A-19. Select a PIR to delete by pressing the box to the left of the corresponding PIR and then pressing the “Delete” button. The PIR will be removed from the database and the new PIR list will be displayed. To remove all PIR hierarchical data, select the “Select All” button. When completed, press the “Done” button to close the window. If the submenu selection ‘Delete’ is erroneously selected, press the “Cancel” button to close the window with no changes to the system.

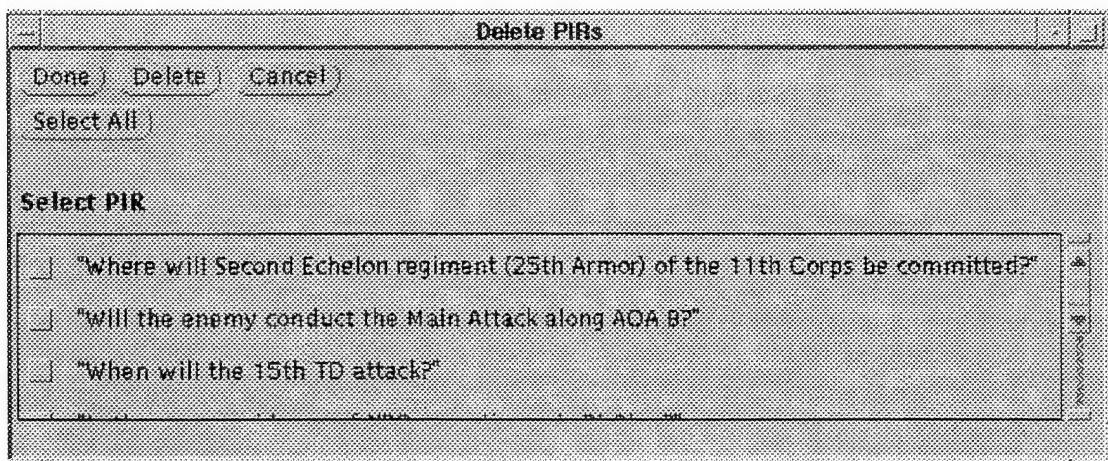


Figure A-19: RM Module - Delete PIRs Window

c. Nodes

To enter new node information into the database or to edit existing node data select the “Nodes” button from the RM module using the right mouse button. A submenu with two choices will be displayed: “Enter New” and “Edit”.

To enter new node information, select the “Enter New” submenu choice using the left mouse button. An “Add Node” window is displayed as in Figure A-20. Enter the Name, Description, and Signature data and select the “Add Node” button. The node will be added to the master list and the “Enter New” window will be cleared for multiple node entry. Press the “Quit” button to close the window.

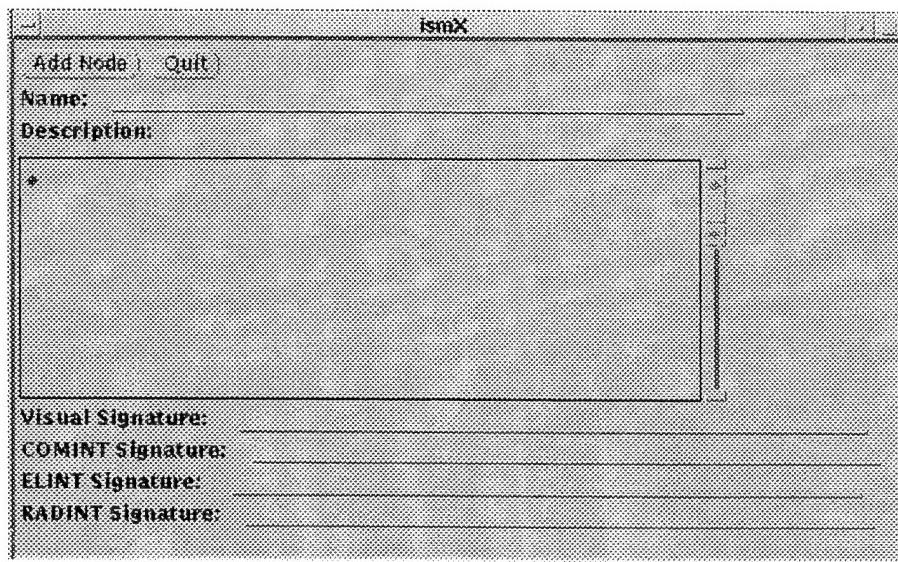


Figure A-20: RM Module - Add New Node Window

To edit node information, select the “Edit” submenu choice using the left mouse button. An “Edit an Existing Node” window is displayed as in Figure A-21. Select a node to edit by pressing the box to the left of the Node text in the choice column.

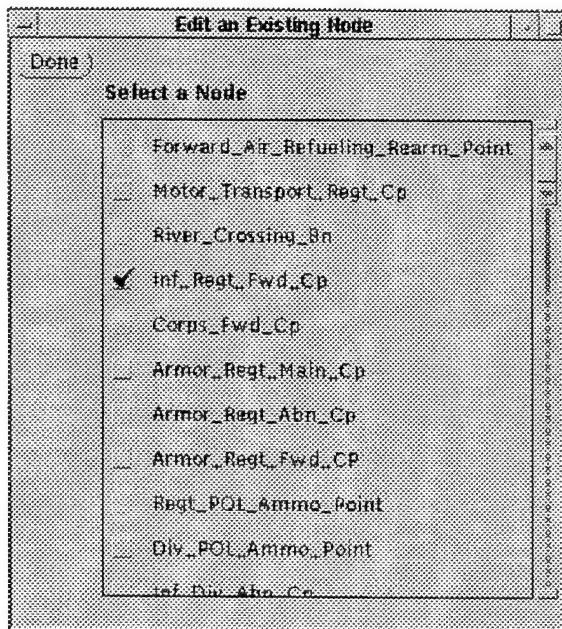


Figure A-21: RM Module - Edit an Existing Node Choice Window

A “Node Information” window is displayed with the node’s data filled in at each prompt. Make the necessary changes and press the “Done” button to save the data. When editing is completed, select the “Quit” button to close the window. If a node is erroneously selected, press the “Quit” button and no changes will be made to the node database.

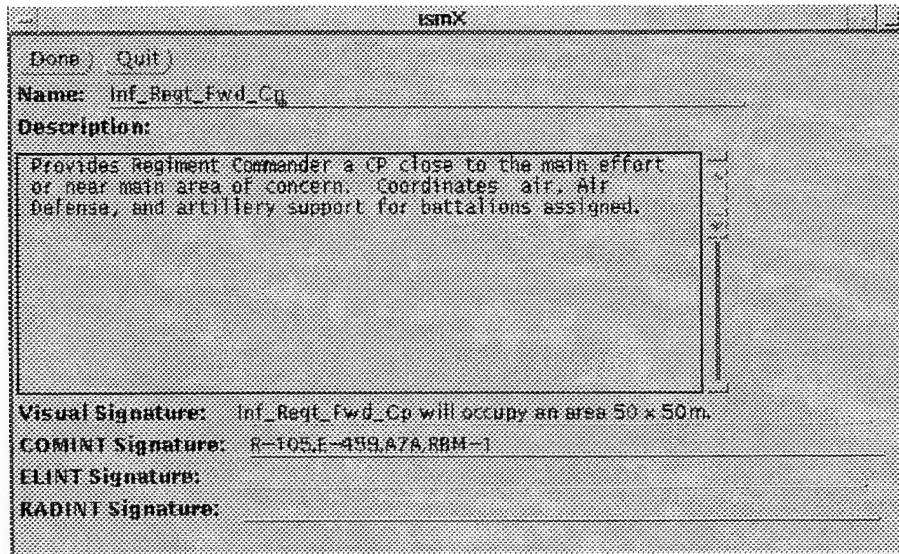


Figure A-22: RM Module - Edit Selected Node Window

3. Defining AO Vulnerabilities

The two primary physical factors that can affect an asset’s collection capability on the battlefield within a specified AO are threat and weather. The user is able to predefine these vulnerabilities that may affect overall asset taskings in a collection plan.

a. Threat Conditions

To predefine the threat vulnerability select the “Threat” button on the RM main menu bar. The ‘Threat Vulnerability’ window that is displayed is shown in Figure A-23. Select the threat vulnerabilities pertaining to an AO by pressing the square box to the right of each threat line with the left mouse button. If the AO has all the threat vulnerabilities, press the “Select All” button. If the “Threat” button is erroneously selected,

press the “Cancel” button and close the window with no changes to the system. When the current threat is defined, select the “Done” button to close the window and save the selections. If no threat vulnerability is defined, the system will use the default threat setting: none.

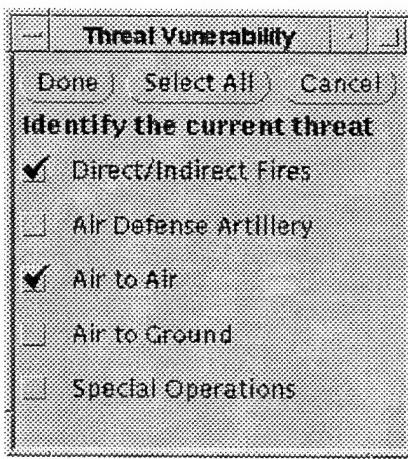


Figure A-23: RM Module - Threat Vulnerability Window

b. Weather Conditions

To predefine the weather conditions select the “Weather Conditions” button on the RM main menu bar. The ‘Weather Forecast’ window that is displayed is shown in Figure A-21. There are eight categories of weather to be defined, each category can be set to one of nine values. For all categories, except Visibility of the Surface which is opposite, the lower values indicate better weather conditions. To select a category setting, use the slide bars to automatically set the value. If the “Weather Conditions” button is erroneously selected, the select the “Cancel” button and close the window with no changes to the system. When the weather is defined, press the “Done” button to close the window and save the selections.

If the weather is not defined, the system will use the default weather setting: cloud cover: 0 - clear; direction of surface winds: 0 - calm; force of surface winds: 0 - calm, < 3 m.p.h.; visibility of the surface: 9 - > 50 km; present weather and obstruction of vision:

0 - no significant weather; state of roads: 0 - dry; state of terrain: 0 - dry; state of water surface: 0 - water level normal.

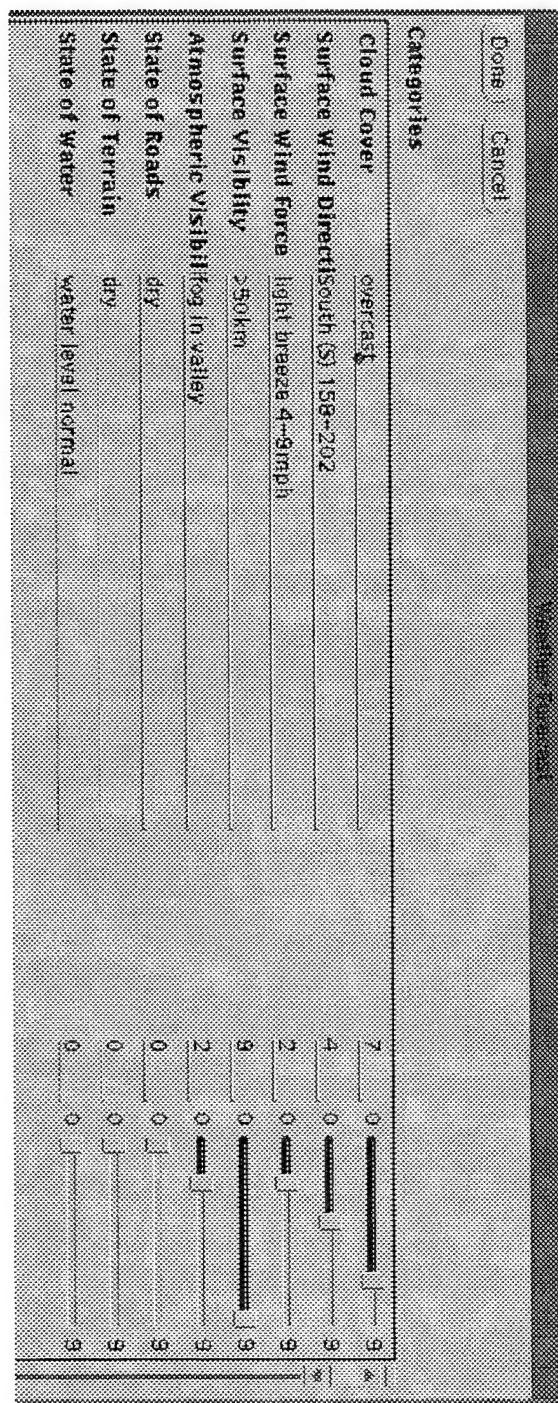


Figure A-24: RM Module - Weather Conditions Window

4. Creating the Collection Plan

Creating the collection plan involves selecting assets and PIRs/SIRs that will support a commander's specific COA. The assets and PIRs can be defined in any order. The evaluating algorithms will be invoked upon the second of the two selected. To create a plan, select the "Create Collection Plan" button from the RM Module main menu bar using the right mouse button. A submenu of two choices will be displayed: "Select Assets" and "Select PIRs". A detailed explanation follows.

a. Selecting Assets

To select assets for a collection plan, select the "Select Assets" submenu choice using the left mouse. A "Select Assets" window will be displayed as in Figure A-25. Select the type and quantity of each assets required for the collection plan. If all the assets in the database are to be used in the plan, press the "Select All" button and select the quantity of each asset type. If the "Select Assets" choice is erroneously selected, press the "Cancel" button to close the window. No changes will be made to the system.

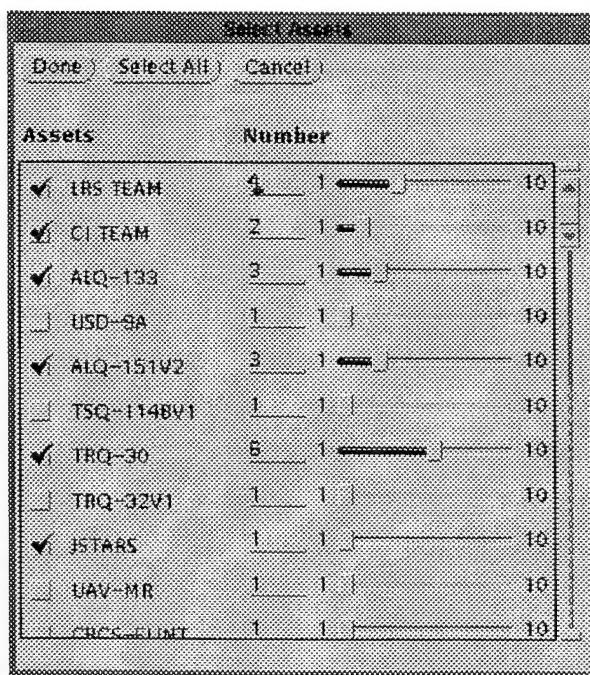


Figure A-25: RM Module - Collection Plan - Select Assets Window

Once the assets have been selected, press the “Done” button and the window in Figure A-26 is displayed. Enter the unique Id Number and Location in UTM grid coordinates for each asset. If the asset selection is not correct, press the “Cancel” button and the window will close and the assets selected will be removed from the collection plan asset list and a new list can be selected from the “Select Assets” window in Figure A-25. Once the data is entered, select the “Done” button and the assets will be displayed on the lower canvas of the RM Module along the y-axis.

Asset Validation		
Plot	Cancel	
Asset	Id Number	Location (UTM)
LRS TEAM	A	AB30004800
LRS TEAM	B	
LRS TEAM	C	
LRS TEAM	D	
CI TEAM	ECHO	
CI TEAM	TANGO	
ALQ-133	W-12	
ALQ-133	W-13	
ALQ-133	W-14	
ALQ-151V2	W-121	
ALQ-151V2	W-122	
ALQ-151V2	W-123	
TRQ-30	A-32	
TRQ-30	A-33	
TRQ-30	A-34	
TRQ-30	A-35	
TRQ-30	A-36	
STARS	THREE	

Figure A-26: RM Module - Collection Plan - Validation Window

b. *Selecting PIRs*

To select PIRs/SIRs for a collection plan, select the “Select PIRs” submenu choice using the left mouse. A “PIR Selection” window will be displayed as in Figure A-27. Select a single PIR and one to many SIRs and press the “Select” button. Repeat this

until all PIRs/SIRs are selected. If the “Select PIRs” choice is erroneously selected, press the “Cancel” button to close the window. No changes will be made to the system.

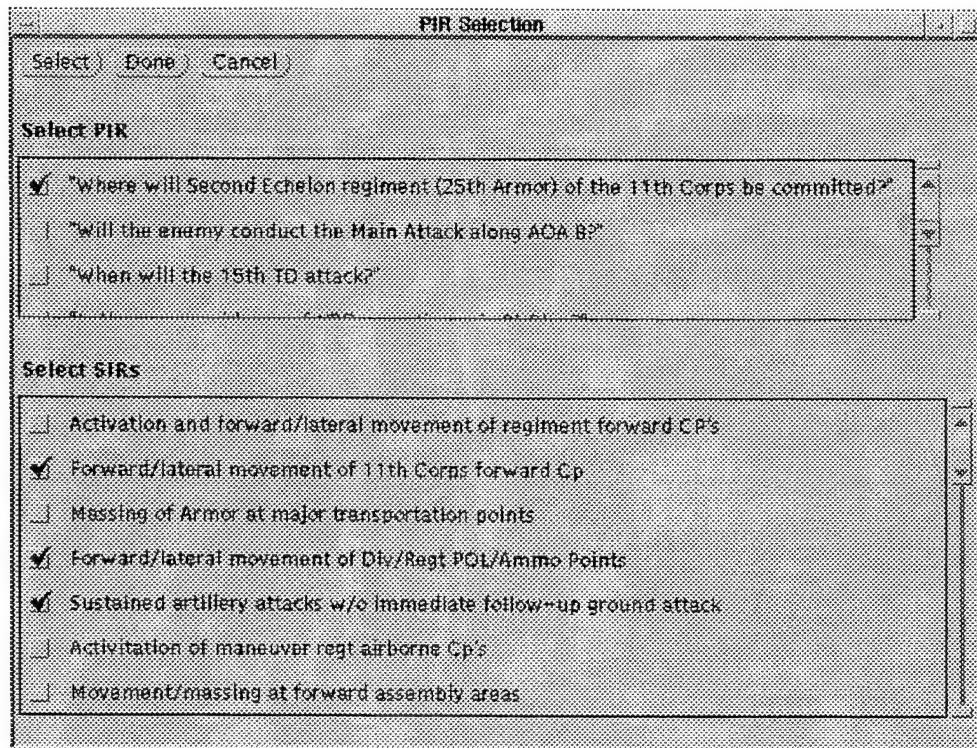


Figure A-27: RM Module - Collection Plan - PIR Selection Window

Once all the PIRs are chosen, press the “Done” button and the window in Figure A-28 is displayed. Select the priority, start time, and end time for each PIR by using the slide bars. The PIR priorities must be unique. After setting the priorities and times, select the “Plot” button and the PIR data will be displayed on the upper and lower canvas of the RM module. The PIR text is displayed on the upper canvas and the PIR priority is displayed across the x-axis.

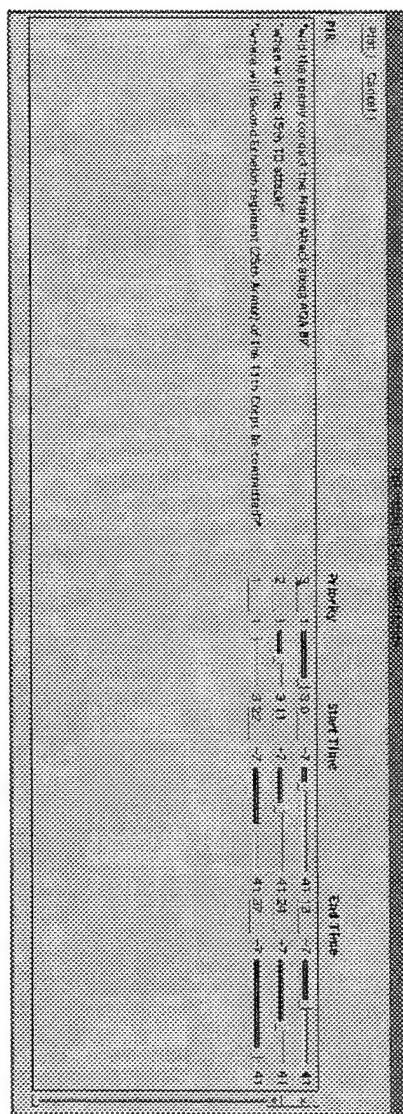


Figure A-28: RM Module - Collection Plan - PIR Validation Window

Figure A-29 shows an example collection plan. In each of the upper and lower canvas' of the RM Module, the screens can be scrolled up and down by either using the slider bar on the right side and lower side of each of the canvas'. An alternate scrolling method is to use the middle mouse button. Select the PIR text in the upper canvas of the RM Module to display the SIR and node information specific to that PIR. To revert back to the PIR text only, select the PIR text again.

Request	Type	Notes	Requirements Management		
			Identify	Resolve	Create Collection Plan
1-1 "Will the enemy conduct the Main Attack along AOA B?"					
2-1 "When will the 15th TRD attack?"					
3-1 "Where will Second Battalion regiment (225th Armer) of the 11th Corps be committed?"					
Collection Assets	PIR	1-1	1-1	1-1	1-1
LRS TEAM	Tasked	Capable	Tasked		
C1 TEAM	Tasked	Capable	Capable		
AfQ 133	NotCapable	NotCapable	NotCapable		
1st Bn 9A	Tasked	NotCapable	NotCapable		
AfQ 13342	Tasked	NotCapable	NotCapable		
TSQ-114B41	Tasked	NotCapable	NotCapable		
TRQ-30	Capable	NotCapable	NotCapable		

Figure A-29: RM Module - Collection Plan

Select the asset in the vertical column of the lower canvas. An ‘Asset Information’ window will be displayed as in Figure A-10. This window will also display the Id Number and UTM grid coordinates.

To view all taskings associated with a PIR, press the PIR priority number in the lower canvas. A ‘PIR/Tasking Information’ window is displayed as in Figure A-30. This is a non-editable window that shows the PIR/SIR, tasking number, asset, and start and end times. Select the “Done” button to close the window.

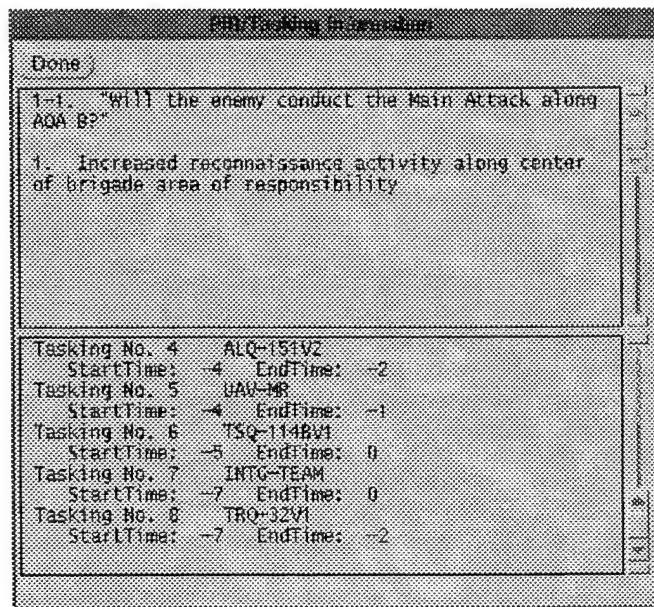


Figure A-30: RM module - PIR/Tasking Information Window

At the intersection of each asset and PIR is a matrix box which provides an evaluation of the asset in reference to its capability of collecting against a PIR. The matrix box contents will contain the following evaluations: Capable, Marginal, and Not Capable. An asset can be tasked for collection by selecting any of the matrix.

Select the “Capable” matrix box, a window is displayed as shown in Figure A-31. Use the slide bars to set the tasking start and end time and press the “Add Tasking” button. Repeat this for each process. The taskings will be displayed in the table. Select the “Done” button when all the taskings have been entered. If the “Capable” is erroneously

pressed, select the “Cancel” button to close the window. No changes will be made to the system.

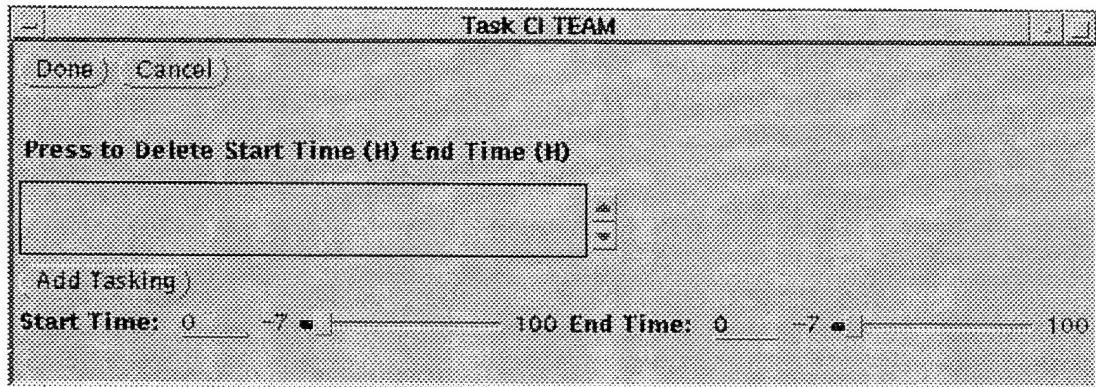


Figure A-31: RM Module - Tasking Window

Select the “Marginal” or “Not Capable” matrix box, an ‘Asset Capability’ window is displayed as shown in Figure A-32. The text in the table gives an explanation for the asset’s collection capability degradation. The capability rating can be overridden by selecting the “Task” button. The ‘Task <asset name>’ window is displayed as in Figure A-31. The same procedure is followed for tasking. To close the ‘Asset Capability’ window, press the “Done” button. If the ‘Asset Capability’ window is erroneously selected, press the “Cancel” button and the window will be closed with no changes to the system.

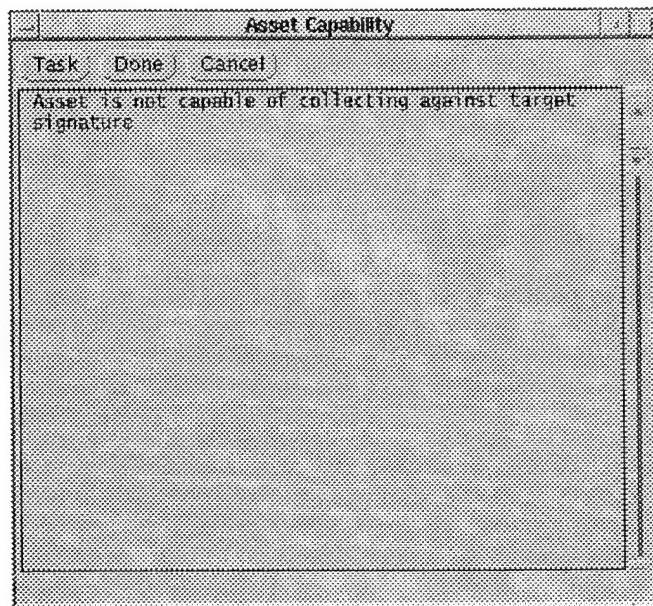


Figure A-32: RM Module - Asset Capability Window

To edit an asset's tasking, select the "Tasked" matrix box. An "Edit Tasking" window will be displayed as in Figure A-33. Taskings can be added as discussed previously or deleted by pressing the choice box to the left of the tasking start and end times.

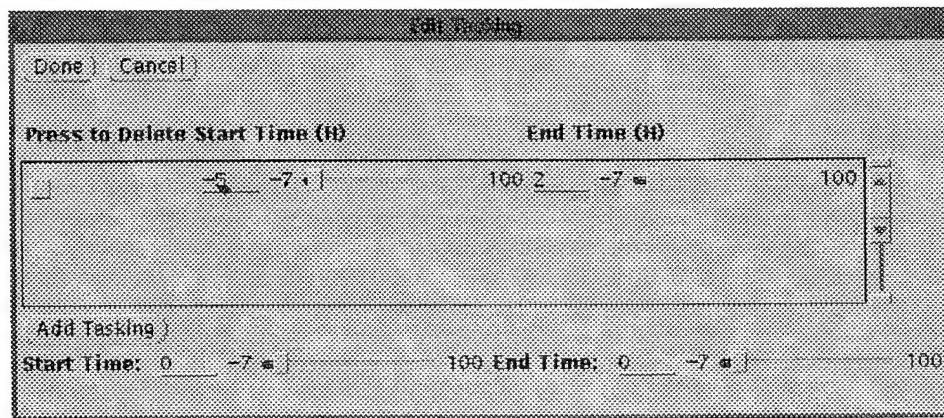


Figure A-33: RM Module - Edit Tasking Window

5. Draw Package

To invoke the drawing package, select the “Draw Package” button from the RM Module main menu bar. A “Zoom Window” will be displayed as in Figure A-34. The line width factor can be scaled by selecting the first down arrow with the left mouse button. The choices are 0.25, 0.5, 1.0, 2.0, and 5.0. The object shapes can be changed by selecting the second down arrow with the left mouse button. The choices are line, polyline, circle, box, and polygon. The line color can be change by selecting the third down arrow with the left mouse button. The color choices are black, white, red, blue, green, yellow. To close the window, select the “Quit” button. Any diagrams drawn will be saved for the current session of the program.

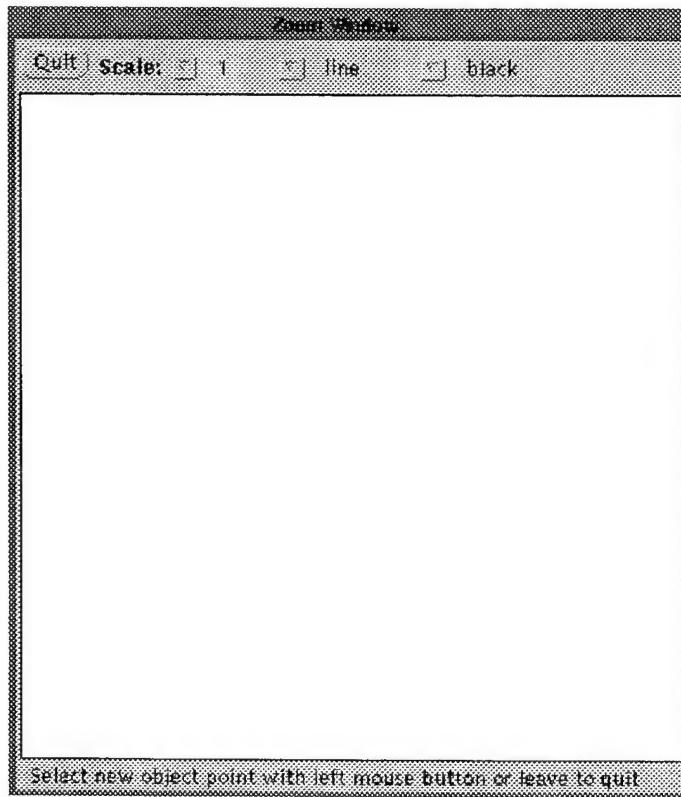


Figure A-34: RM Module - Zoom Window

6. Help

To get help or information about the matrices, select the “Help” button with the right mouse button. A submenu with the different synchronization matrices will be displayed as in Figure A-5. Select the synchronization matrix of choice with the left mouse button. The WWW browser, Mosaic, will be invoked and display the help information.

7. Exiting to ISM Matrix

Upon completion of a collection plan, select the “IEW Sync Matrix” button on the main menu bar of the RM module. The AM module screen is displayed and the current plan is loaded. An example is shown in Figure A-35. The canvas shows a matrix of assets vs. time with all the taskings and schedules plotted. This is a working document and can be edited to meet the battlefield situation.

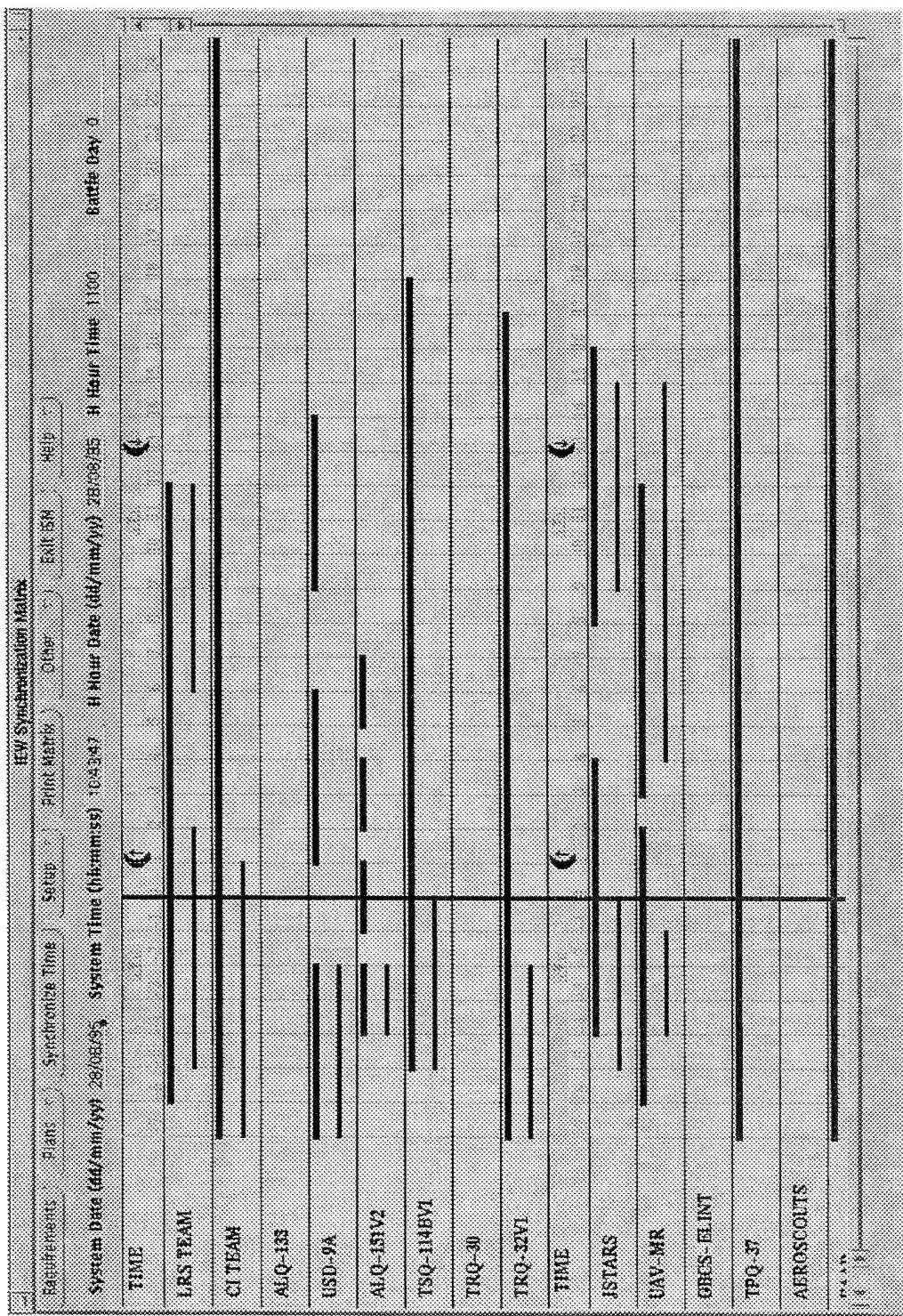


Figure A-35: AM Module - Asset Management Plan

APPENDIX B. WEATHER FORECAST INFORMATION

This appendix contains the weather condition information used in the requirements management module for evaluating AO vulnerabilities. There are eight categories of weather settings each with nine option settings. The following is a breakout the weather categories:

- Cloud Cover (Setting and Description)
 - 0 clear
 - 1 None
 - 2 Scattered
 - 3 Scattered (hills in clouds)
 - 4 None
 - 5 Broken
 - 6 Broken (hills in clouds)
 - 7 Overcast
 - 8 Overcast (hills in clouds)
 - 9 None
- Direction of Surface Winds
 - 0 Calm None
 - 1 Northeast (NE) 023 to 067
 - 2 East (E) 068 to 112
 - 3 Southeast (SE) 113 to 157
 - 4 South (S) 158 to 202
 - 5 Southwest (SW) 203 to 247
 - 6 West (W) 248 to 292
 - 7 Northwest (NW) 293 to 337
 - 8 North (N) 338 to 022

- 9 Variable None
- Force of Surface Winds
 - 0 Calm <3 m.p.h.
 - 1 --
 - 2 Light Breeze 4-9 m.p.h.
 - 3 --
 - 4 Moderate Breeze 10-19 m.p.h.
 - 5 --
 - 6 Strong Breeze 20-29 m.p.h.
 - 7 --
 - 8 Gale >30 m.p.h.
 - 9 --
- Visibility of the Surface
 - 0 <164ft (<50m)
 - 1 165 to <656 ft. (50 to <200m)
 - 2 656 to <1640 ft. (200 to 500m)
 - 3 1640 to <3280 ft. (500 to <1000m)
 - 4 3280 ft. to <1.2 mi (1 to <2km)
 - 5 1.2 to <2.48 mi (2 to <4km)
 - 6 2.48 to <6.21 mi (4 to <10km)
 - 7 6.21 to <12.42 mi (10 to <20km)
 - 8 12.42 to <31.06 mi (20 to <50km)
 - 9 >31.06 mi (>50km)
- Present Weather and Obstruction of Vision
 - 0 No significant weather
 - 1 Smoke or haze
 - 2 Fog in valley
 - 3 Sandstorm, dust storm, or blowing snow

- 4 Fog
- 5 Drizzle
- 6 Rain
- 7 Snow or rain and snow mixed
- 8 Showers
- 9 Thunderstorms with or without precipitation
- State of Roads
 - 0 Dry
 - 1 Wet
 - 2 Flooded
 - 3 Slush
 - 4 Ice Patches
 - 5 Glazed Ice
 - 6 Snow Depth 0 to 7.48 in (0 to 19cm)
 - 7 Snow Depth 1.97 to 9.45 in (>20cm)
 - 8 Snow Drift
 - 9 --
- State of Terrain
 - 0 Dry
 - 1 Wet
 - 2 Pools of Water on Surface
 - 3 Flooded
 - 4 Ground Frozen 0 to 1.5 in (0 to 4cm)
 - 5 Ground Frozen 1.97 to 9.45 in (>5cm)
 - 6 Snow Depth 0 to 1.5 in (0 to 4cm)
 - 7 Snow Depth 1.97 to 9.45 in (5 to 24cm)
 - 8 Snow Depth 9.45 to 17.32 in (25 to 44cm)
 - 9 Snow Depth >17.32 in (>45cm)

- State of Water Surface
 - 0 Water Level Normal
 - 1 Water Level Much Below Normal
 - 2 Water Level High, but Not Overflowing
 - 3 Banks Overflowing
 - 4 Floating Ice (> then half)
 - 5 Thin Ice 0 to 1.5 in (0 to 4 cm) thick, complete cover
 - 6 Ice Depth unknown, complete cover, passable for persons
 - 7 Ice Depth 1.97 to 3.54 in (5 to 9 cm)
 - 8 Ice Depth 3.93 to 9.44 in (10 to 24 cm)
 - 9 Ice Depth >9.48 in (>25 cm)

LIST OF REFERENCES

- [ASAS95] All Source Analysis System Program Office, “*All Source Analysis System Server*”, <http://www.army.mil/pmif-pg/asas.htm>, July, 1995.
- [BCBL95] Battle Command Battle Lab, “*Proposal for the U. S. Army Artificial Intelligence Center*”, Battle Command Battle Technology Lab, Fort Huachuca, Arizona, August, 1995
- [BCBL94] Battle Command Battle Lab, “*Battle Command Decision Support System Development Brief*”, Battle Command Battle Technology Lab, Fort Huachuca, Arizona, October, 1994.
- [FUJA94] Fuja, R.S. and Widing, M.A., “*Application Interface Engine Language and Object Reference Manual*”, Advanced Modeling and Analysis Section, Environmental Assessment and Information Sciences Division, Argonne National Laboratory, Argonne, Illinois, April, 1994.
- [FM34-1] Department of the Army, *FM 34-1, “Intelligence and Electronic Warfare Operations”*, Government Printing Office, Washington DC, June, 1994.
- [FM34-2] Department of the Army, *FM 34-2, “Collection Management and Synchronization Planning”*, Government Printing Office, Washington DC, March, 1994.
- [FM34-81] Department of the Army, *FM 34-81, “Weather Support for Army Tactical Operations”*, Government Printing Office, Washington DC, August, 1989.
- [FM34-130] Department of the Army, *FM 34-130, “Intelligence Preparation of the Battlefield”*, Government Printing Office, Washington DC, July, 1994.
- [ST100-9] U.S. Army Command and General Staff College, *Student Text 100-9, “Techniques and Procedures for Tactical Decisionmaking”*, Fort Leavenworth, Kansas, July, 1991.
- [WAYNE90] Major General Wayne Knudson, “The Future of C²”, Military Review, pp. 5-16, November, 1993.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
Dudley Knox Library Code 013 Naval Postgraduate School Monterey, CA 93943-5101	2
Chairman, Code CS Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Dr David R. Pratt, Code CS/PR Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Dr Man-Tak Shing, Code CS/S Computer Science Department Naval Postgraduate School Monterey, CA 93943	1
CPT Karen D. H. Tomlin 231 Metz Road Seaside, CA 93955	1